

BAB IV

HASIL DAN PEMBAHASAN

4.1 *Pre-Production*

Tahap *Pre-Production* adalah tahap awal dari semua proses yang dilakukan, tahap ini adalah tahap untuk mendefinisikan dan melakukan perencanaan terhadap permainan yang akan dibangun. Tahap *Pre-Production* sudah dilakukan pada penelitian sebelumnya oleh Muhammad Ilham Nur Isra' [8]. Pada penelitian tersebut menghasilkan *Game Design Document*(GDD) dan *Software Requirements Specification*(SRS) untuk permainan ular tangga bergenre strategi. penulis hanya menjelaskan secara singkat hasil dari penelitian yang dilakukan sebelumnya. Untuk melihat versi lengkap GDD dan SRS yang dibuat dapat dilihat pada penelitian sebelumnya.

4.1.1 *Game Design Document (GDD)*

Secara singkat, permainan yang dibuat adalah permainan ular tangga yang diperbaharui dengan menambahkan unsur-unsur strategi di dalamnya. Strategi yang di-*design* pada permainan ini adalah *Turn Based Strategy* dengan *platform* yang digunakan adalah *dekstop*. Target dari pengguna permainan ini adalah orang dewasa dengan kisaran umur 20-35 tahun.

Permainan ini memiliki beberapa daya tarik. Pertama, berbeda dengan permainan ular tangga pada umumnya, permainan ini dimulai dengan keadaan *board* yang kosong tanpa adanya ular dan tangga. Kedua, penggunaan kartu untuk mendapatkan ular dan tangga. Ketiga, adanya penggunaan poin sebagai alat untuk jual beli kartu. Keempat, adanya fitur simpan kartu. Kelima, selain ular dan tangga ada penambahan item baru seperti Bom, Ular Kepala 2, Tangga Kaki 2, dan lain-lain.

Untuk *mechanic* dalam permainan ular tangga bergenre strategi ini terdiri dari 2 hal yaitu elemen dan *rule*. Elemen adalah unsur-unsur yang dibutuhkan dalam permainan ini sedangkan *rule* adalah aturan yang digunakan dalam permainan ini.

Ada 11 elemen yang digunakan pada permainan ini. Elemen-elemen yang digunakan dapat dilihat pada tabel 4.1.

Tabel 4.1 Elemen yang digunakan pada permainan ular tangga bergenre strategi

No.	Elemen	Keterangan
1	Kotak <i>start</i>	Kotak <i>start</i> adalah kotak yang digunakan sebagai area pion untuk memulai permainan
2	<i>Board</i>	<i>Board</i> /papan permainan merupakan arena yang digunakan pemain untuk menjalankan pion. Board berisikan 100 kotak
3	Kotak <i>Finish</i>	Kotak <i>finish</i> berada pada kotak no. 100, kotak ini merupakan kotak yang dituju untuk memenangkan permainan
4	Dadu	Dadu merupakan alat yang digunakan untuk menentukan jumlah langkah pemain dan jumlah poin yang ditambahkan
5	Poin	Poin merupakan alat yang digunakan pemain untuk membeli kartu. Poin dapat diperoleh dari melempar dadu
6	Pion	Pion merupakan karakter pemain yang digunakan untuk mewakili pemain dalam menjalankan permainan

No.	Elemen	Keterangan
7	Poin Bar	Poin bar merupakan sebuah wadah yang digunakan untuk mengumpulkan jumlah poin milik pemain
8	<i>Card Shop</i>	Card shop adalah sebuah area yang menyediakan kartu yang dapat dibeli oleh pemain. berisi kartu Level 1, kartu Level 2 dan kartu Level 3
9	<i>Save Area</i>	Save Area adalah area untuk menyimpan kartu. Jumlah total slot kartu yang dapat disimpan adalah 3 kartu
10	Status Area	Area yang menunjukkan status pemain yang sedang bermain
11	<i>Card On Hand</i>	Area yang menunjukkan kartu yang ada ditangan. Ketika kartu ada di <i>area Card On Hand</i> pemain dapat melakukan <i>Sell/Jual</i> , <i>Save/Simpan</i> , <i>Use/Pakai</i> kartu
12	Kartu	Kartu digunakan pemain untuk menentukan <i>item</i> yang didapat. Kartu dapat dibeli menggunakan poin yang dimiliki oleh pemain.
13	<i>Item</i>	<i>Item</i> merupakan alat yang digunakan pemain sebagai alat bantu untuk menjalankan strategi selama permainan berlangsung. <i>Item</i> didapat berdasarkan kartu yang didapat. Item dapat dikenakan langsung ke pion lawan, atau memasang pada board.

Untuk elemen kartu dan item memiliki 3 level yaitu level 1, level 2, dan level 3. Untuk lebih jelasnya mengenai jumlah dan perinciannya dapat dilihat pada tabel 4.2.

Tabel 4.2 Daftar Kartu dan *Item* [8]

Level Kartu	Item	Rincian	Jumlah	
Level 1	Ular Level 1	↓	2	
		↓, ← ₂	2	
		↓, → ₂	2	
		↓ ₂	2	
		↓ ₂ , ←	2	
		↓ ₂ , →	2	
	Tangga Level 1	↑	2	
		↑, ← ₂	2	
		↑, → ₂	2	
		↑ ₂	2	
		↑ ₂ , ←	2	
		↑ ₂ , →	2	
	Total			24
	Level 2	Ular Level 2	↓ ₃	2
↓ ₃ , ←			2	
↓ ₃ , →			2	
↓ ₃ , ← ₂			1	
↓ ₃ , → ₂			1	
Tangga Level 2		↑ ₃	2	
		↑ ₃ , ←	2	

Level Kartu	Item	Rincian	Jumlah	
		\uparrow_3, \rightarrow	2	
		\uparrow_3, \leftarrow_2	1	
		$\uparrow_3, \rightarrow_2$	1	
		Ular Kepala 2	\downarrow_3	2
		Tangga Kaki 2	\uparrow_3	2
		Steal	Mencuri 1 item lawan	2
		Bom M	Mundur 5 langkah	2
Total			24	
Level 3	Ular Level 3	\downarrow_4	1	
		\downarrow_4, \leftarrow	1	
		$\downarrow_4, \rightarrow$	1	
		$\downarrow_4, \leftarrow_2$	1	
		$\downarrow_4, \rightarrow_2$	1	
		$\downarrow_4, \leftarrow_3$	1	
		$\downarrow_4, \rightarrow_3$	1	
		\downarrow_5, \leftarrow	1	
		$\downarrow_5, \rightarrow$	1	
	Tangga Level 3	\uparrow_4	1	
		\uparrow_4, \leftarrow	1	
		\uparrow_4, \rightarrow	1	
		\uparrow_4, \leftarrow_2	1	
		$\uparrow_4, \rightarrow_2$	1	
		\uparrow_4, \leftarrow_3	1	

Level Kartu	Item	Rincian	Jumlah
		↑ ₄ , → ₃	1
		↑ ₅ , ←	1
		↑ ₅ , →	1
	Bom L	Mundur 10 langkah	2
	Stun	Dilarang main 1 turn/giliran	2
	Stealth	Menyamar pion lawan 2 turn/giliran	2
Total			24
Keterangan : ↑ : ke atas ↓ : ke bawah 1, 2, 3.... : jumlah langkah → : ke kanan ← : ke kiri			

Pada bagian *rule* berisikan fase-fase yang dibagi ke dalam 2 kategori, *preparation game* dan *in game*. Untuk perinciannya adalah sebagai berikut:

A. Preparation Game

1. *Starting player*, adalah fase yang digunakan untuk menentukan urutan main pemain.
2. *Starting Game*, adalah tahap mempersiapkan kondisi *game* sebelum *game* dimulai. hal yang perlu disiapkan adalah posisi pion milik setiap *player* berada di kotak start, Keadaan *board* kosong, tanpa ada *item* didalamnya, kondisi pion bar pada setiap *player* dalam keadaan kosong tanpa pion, kondisi kartu pada tiap level dalam posisi teracak, dan *player* akan bermain secara berurutan sesuai dengan urutan yang didapat dari *Starting Player*.

B. In Game

1. *Turn phase*, adalah fase ketika pemain melakukan *roll* dadu dan menjalankan pion.
2. *Strategy phase*, adalah fase yang terjadi setelah *turn phase*. Fase ini adalah fase dimana pemain dapat melakukan strategi yang akan dilakukan. dalam fase ini ada 5 hal yang bisa dilakukan oleh pemain.
 - a) *Buy/*beli, pemain dapat membeli kartu dengan poin yang dimiliki dengan harga:
 - Level 1 : 7 poin.
 - Level 2 : 10 poin.
 - Level 3 : 13 poin.
 - b) *Pakai/*use, pemain dapat memakai kartu yang sudah diperoleh, dan akan ditukar dengan item yang sesuai dengan keterangan pada kartu.
 - c) *Simpan/*save, pemain dapat menyimpan kartu yang diperoleh selama permainan. Kapasitas simpan kartu tiap pemain adalah 3 slot. Kartu hasil penyimpanan hanya bisa dipasang ketika *strategy phase*. Kartu yang disimpan akan masuk kedalam *Save Area*.
 - d) *Jual/*sell, pemain dapat menjual kartu yang ia miliki dengan harga:
 - Level 1 : 4 poin.
 - Level 2 : 6 poin.
 - Level 3 : 8 poin.
 - e) *End*, dimana pemain mengakhiri gilirannya. End adalah pertanda bahwa giliran pemain sudah berakhir. Dan akan dilanjutkan oleh pemain dengan urutan berikutnya.

Untuk *flow* dari permainan ular tangga bergenre strategi ini dimulai dari *Preparation Game* yang dimana pemain menghadapi fase *starting player* dan *starting game*. Setelah tahap *preparation game*, dilanjutkan ketahap *In Game*. Pada tahap ini pemain akan menghadapi 2 fase, *turn phase* dan *strategy phase*. Pemain melakukan *turn phase* terlebih dahulu dan setelah itu melakukan *strategy phase*.

Sedangkan untuk *controller*-nya sendiri, karena target *platform* pada permainan ini adalah *desktop*, untuk *control* yang digunakan pada permainan ini menggunakan *mouse click (Left Click)*.

Dengan dibuatnya GDD ini diharapkan pemain akan merasa senang ketika bisa mengalahkan lawannya. Dan pada pihak lawan akan merasa sedih atau kesal ketika tidak bisa memenangkan *game* ini. Dan akan memberikan pengalaman untuk saling bersaing, saling menjatuhkan satu sama lain untuk memperoleh kemenangan.

4.1.2 Usecase dan Activity Diagram

Pada pembuatan permainan ular tangga bergenre strategi pada penelitian sebelumnya telah menghasilkan *Software Requirements Specification*(SRS). SRS tersebut berupa *usecase* diagram dan *activity* diagram. Untuk *usecase* dan *activity* diagram yang dihasilkan adalah sebagai berikut:

a. Usecase Diagram



Gambar 4.1 Usecase diagram sistem permainan ular tangga bergenre strategi [8]

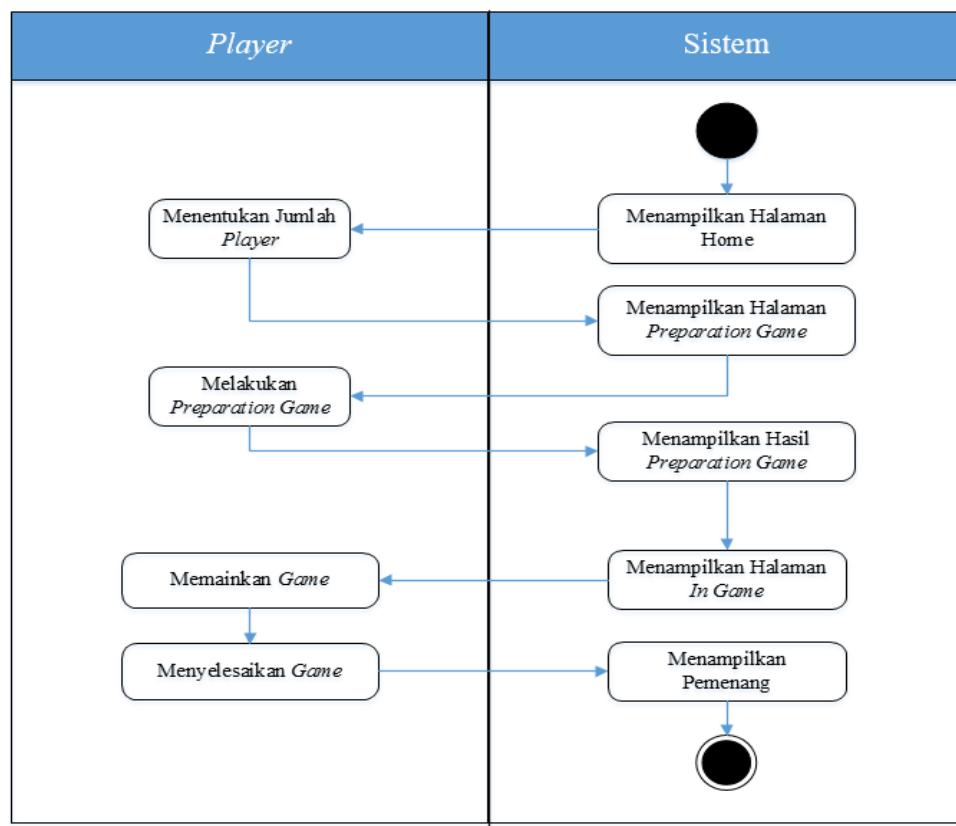
Gambar 4.1 merupakan gambar *Usecase* diagram sistem permainan ular tangga bergenre strategi. Gambar tersebut menjelaskan yang berperan sebagai aktor adalah seorang *player*/pemain. Ada 3 *case* pada sistem permainan ular tangga bergenre strategi ini. *Case* pertama adalah memilih

jumlah *player*. Ada tiga mode pilihan yang dapat dipilih *player* yaitu mode 2 *player*, 3 *player* atau 4 *player*. *Case* kedua adalah melakukan *Preparation Game*, *case* ini digunakan *player* untuk menentukan urutan main *player* ketika bermain. Sebelum melakukan *Preparation Game* ada *include* untuk memilih jumlah *player* terlebih dahulu, jadi *player* tidak bisa melakukan *Preparation Game* sebelum jumlah *player* ditentukan. *Use case* ketiga adalah memainkan *game*, pada bagian ini *player* akan memainkan *game*-nya. Namun sebelum *player* memainkan *game* ada *include* juga yang berarti melakukan *Preparation Game* terlebih dahulu untuk memainkan *game*-nya.

b. Activity Diagram

Ada 8 *activity* diagram dalam permainan ular tangga bergenre strategi ini. berikut ini adalah 8 *activity* diagram tersebut.

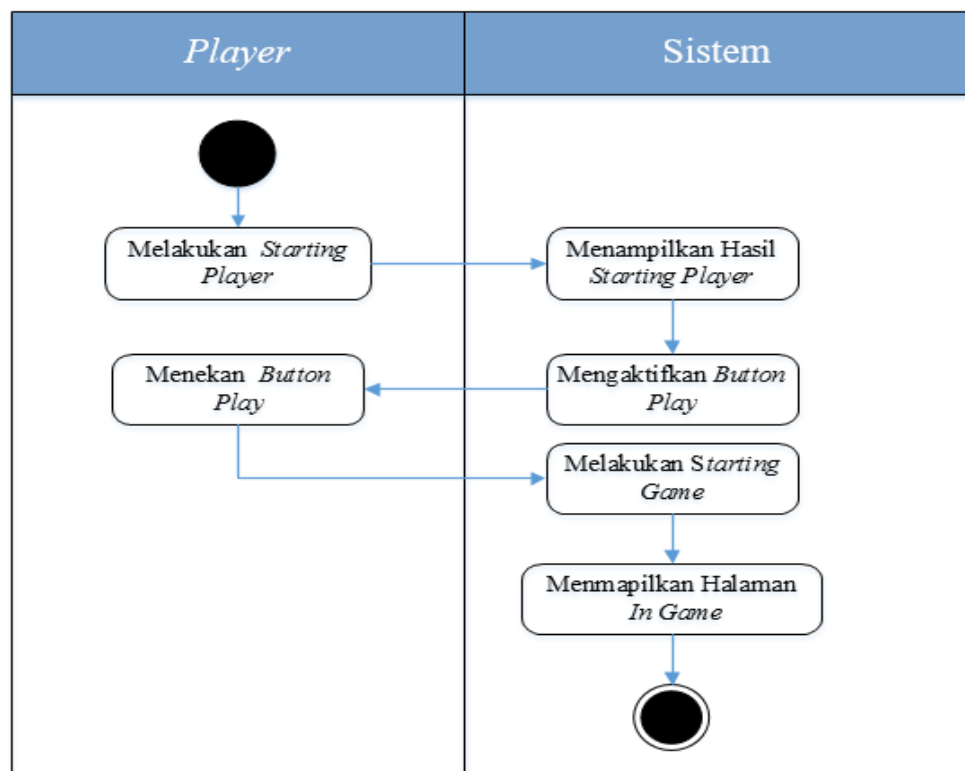
1. Activity Diagram Alur Permainan Ular Tangga Bergnere Strategi



Gambar 4.2 Activity Diagram Alur Permainan Ular Tangga Bergnere Strategi [8]

Gambar 4.2 merupakan gambar *activity* diagram permainan ular Tangga Bergenre Strategi, Gambar tersebut menjelaskan *activity* dimulai dari sistem yang memunculkan halaman home. Setelah halaman Home muncul, *player* akan menentukan jumlah *player* yang bermain. Ada 3 pilihan untuk menentukan jumlah *player* yaitu 2 *player*, 3 *player* atau 4 *player*. Setelah menentukan jumlah *player*, sistem akan menampilkan halaman *preparation game*. Didalam halaman *preparation game*, *player* akan melakukan *preparation game*. Untuk penjelasan *preparation game* dijelaskan pada *activity preparation game*. Setelah *preparation game* selesai, *player* akan memainkan *game* hingga salah satu *player* memenangkan *game*.

2. Activity Diagram Preparation Game

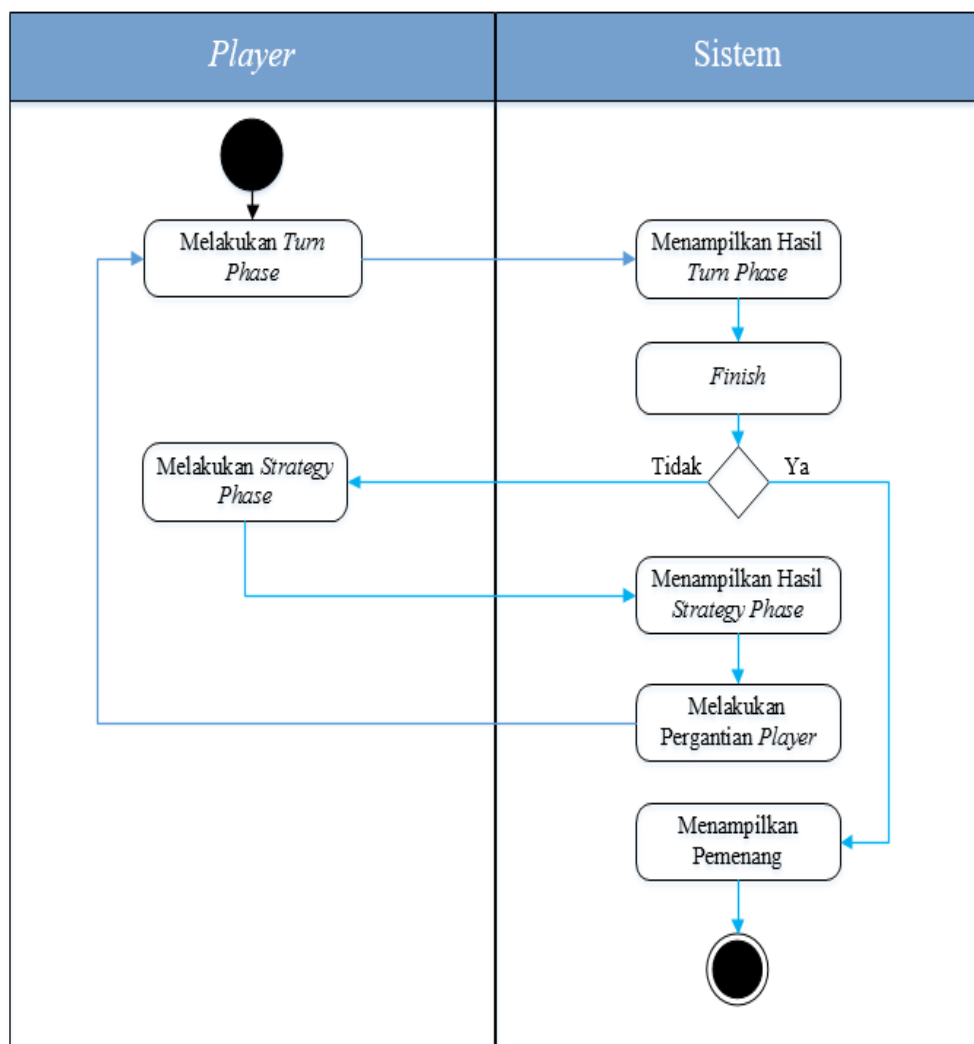


Gambar 4.3 Activity Diagram Preparation Game [8]

Gambar 4.3 merupakan gambar *activity* diagram *Preparation Game*, Gambar tersebut menjelaskan *activity* dimulai dari salah satu *player* melakukan *starting player* untuk menentukan urutan main tiap *player*.

setelah itu, sistem memunculkan hasil dari *starting player*. Hasil tersebut berisi angka giliran yang didapat. Setelah hasil dari *starting player* muncul, sistem mengaktifkan *button Play* untuk menuju ke *game*-nya. Setelah *button Play* aktif, *player* akan menekan *button Play*. Setelah *button Play* ditekan, sistem akan melakukan *starting game* yaitu mempersiapkan permainannya. Hal-hal yang harus dipersiapkan sistem dapat dilihat pada bab 4 poin 4.1.1. Setelah *starting game* selesai dilakukan, sistem menampilkan halaman *in game/permainan*.

3. Activity Diagram In Game



Gambar 4.4 Activity Diagram in game [8]

Gambar 4.4 merupakan gambar *activity diagram In Game*, Gambar tersebut menjelaskan *activity* dimulai dari *player* melakukan *Turn phase*. Pada bagian *Turn Phase* dijelaskan pada *activity Turn Phase*. Setelah *Turn phase* dilakukan, sistem akan memeriksa apakah pion sudah sampai finish, Jika “Ya” maka permainan berakhir dan sistem akan menampilkan pemenang dari *game*. Jika “Tidak”, maka *player* akan melakukan *Strategy phase*, Pada bagian *Strategy Phase*, ada 6 tindakan yang dapat dilakukan *player*, beli kartu, ambil kartu pada *save area*, pakai kartu, jual kartu, simpan kartu dan *End*. *End* merupakan tindakan untuk mengakhiri *Strategy Phase*. Setelah *player* melakukan *End*, sistem akan melakukan pergantian *player* sesuai urutan main dan akan memulai *Turn phase* lagi dari awal.

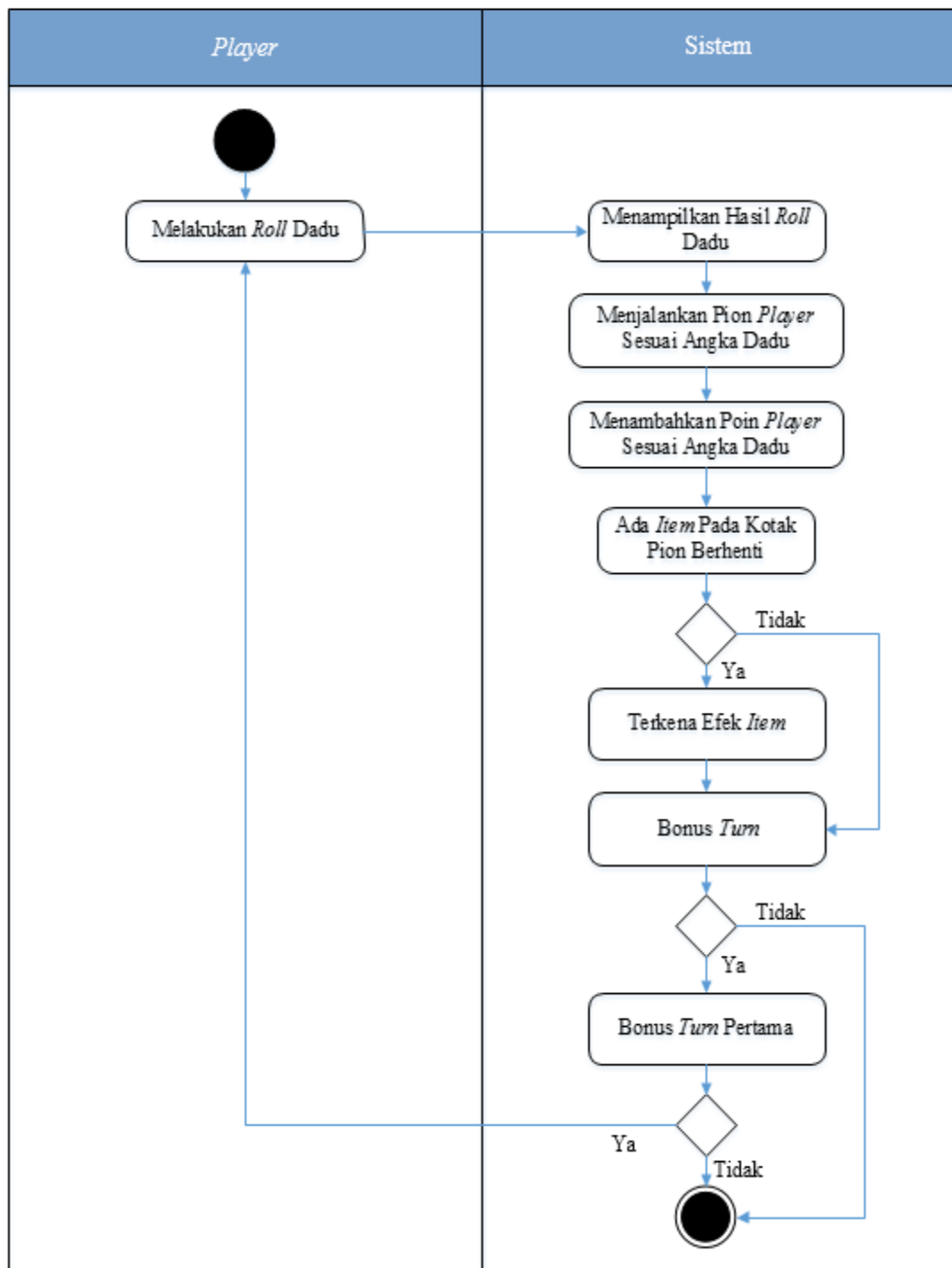
4. Activity Diagram *Turn phase*

Activity diagram *turn phase* dapat dilihat pada gambar 4.5, pada gambar tersebut menjelaskan *activity diagram* dimulai dari *player* melakukan roll dadu untuk menentukan jumlah langkah yang didapat *player*. Selanjutnya sistem akan menjalankan pion dan menambah poin milik *player* sesuai angka jumlah langkah yang didapat *player*.

Setelah pion selesai berjalan, sistem akan melakukan pemeriksaan pada kotak tempat pion berhenti ada item atau tidak. jika “Tidak” sistem akan langsung memeriksa bonus *turn*. Namun jika “Ya” dan kotak tersebut berisi kepala ular milik lawan, maka pion akan turun menuju kotak yang berisi ekor ular. Jika “Ya” dan kotak tersebut berisi kepala ular milik sendiri, maka pion tidak akan turun. Jika “Ya” dan kotak berisi kaki tangga sendiri pion akan naik menuju kepala/ujung tangga. Jika “Ya” dan kotak berisi kaki tangga milik lawan maka pion tidak akan naik.

Setelah pemeriksaan pada kotak tempat pion berhenti selesai dan menjalankan efek seperti yang ditentukan, sistem akan memeriksa bonus *turn*. Bonus *turn* merupakan ketika angka roll dadu *player* mendapat 6. Jika *player* tidak mendapat bonus *turn* maka *Turn phase* langsung berakhir. Jika *player* mendapat bonus *turn*, sistem akan melakukan pemeriksaan kondisi apakah bonus *turn* yang didapat merupakan bonus *turn* yang pertama, jika

“Ya” maka *player* dapat melakukan roll dadu kembali. Jika “Tidak” maka *Turn phase* berakhir.



Gambar 4.5 Activity Diagram *Turn phase* [8]

5. Activity Diagram *Strategy phase*

Activity diagram *strategy phase* dapat dilihat pada gambar 4.6, pada gambar tersebut menjelaskan Ada 6 tindakan yang dapat dilakukan *player* pada bagian *Strategy phase* yaitu beli kartu, ambil kartu pada *save area*, pakai kartu, jual kartu, simpan kartu dan *End*.

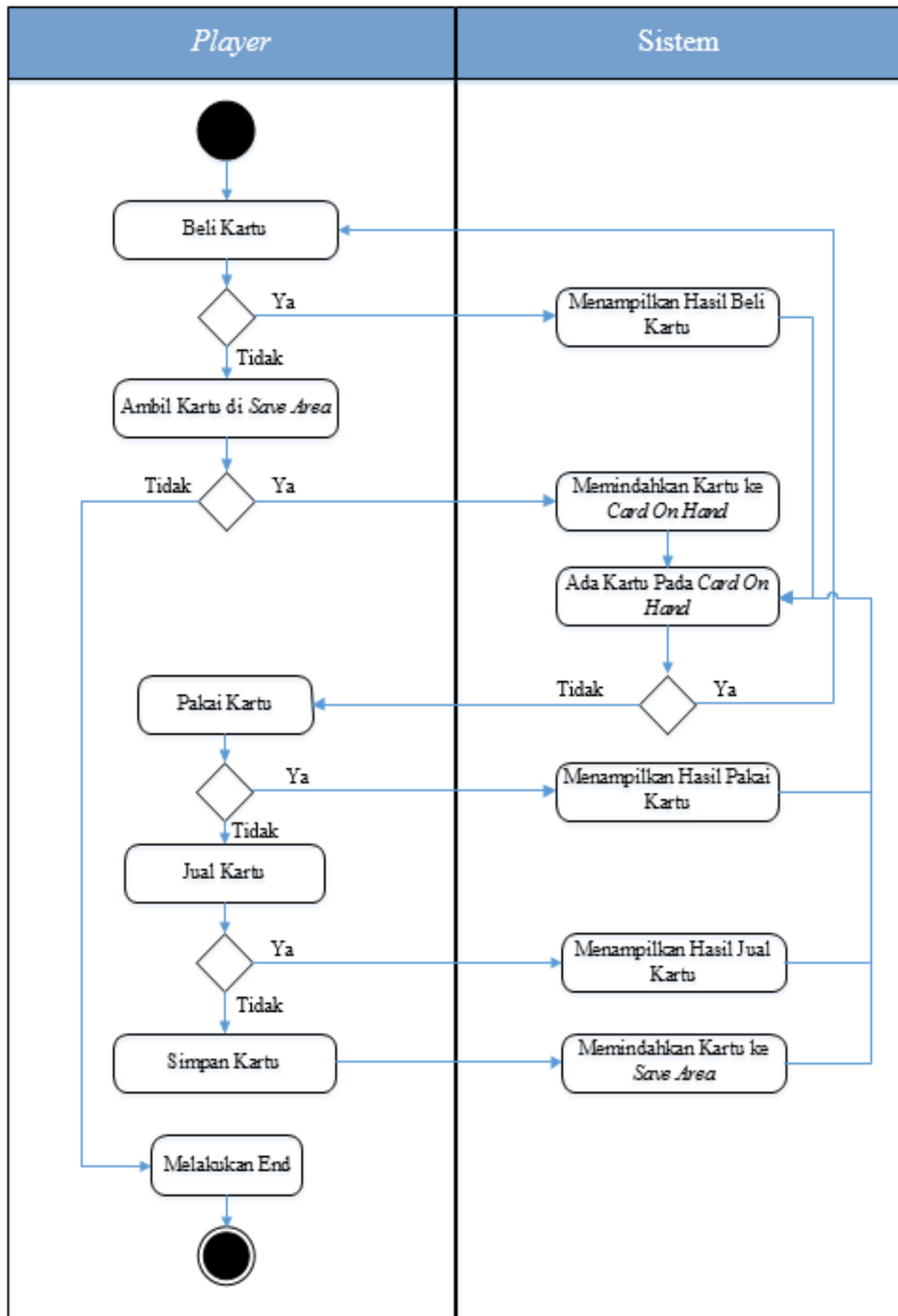
Pada bagian pembelian kartu, *player* hanya bisa membeli kartu ketika poin milik *player* mencukupi harga beli kartu. Untuk penjelasan lebih lanjut tentang Beli kartu dijelaskan pada *activity* beli kartu. Ketika *player* ingin mengambil kartu pada *save area*, *player* hanya bisa mengambil kartu ketika bagian *save area* ada kartunya. Ketika *player* memakai kartu, kartu tersebut akan dikonversi menjadi item, untuk penjelasan lebih lanjut tentang pakai kartu dijelaskan pada *activity* pakai kartu. Ketika *player* menjual kartu, *player* akan mendapat poin dari hasil penjualan sesuai harga jual kartu, untuk penjelasan lebih lanjut tentang jual kartu dijelaskan pada *activity* jual kartu. Ketika *player* melakukan *save*, bagian *save area* minimal harus ada satu bagian yang kosong. Ketika berhasil melakukan *save* kartu, maka kartu yang ada pada *card on hand* akan dipindah kedalam *save area*. *Player* dapat melakukan *End* untuk mengakhiri *strategy phase*.

6. Activity Diagram *Buy/Beli Kartu*

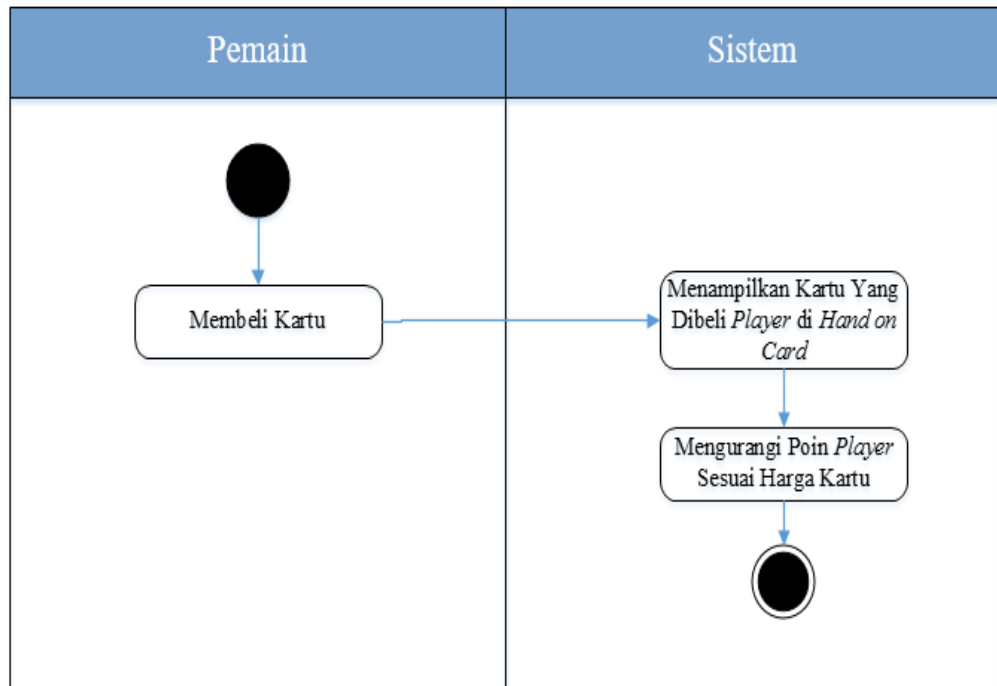
Activity diagram *Buy/Beli Kartu* dapat dilihat pada gambar 4.7, pada gambar tersebut menjelaskan *activity diagram* dimulai dari *player* melakukan beli kartu. Setelah kartu terbeli, sistem akan menampilkan kartu yang didapat pada *Card on Hand*. Lalu, sistem akan mengurangi poin milik *player* sesuai harga kartu. Setelah itu *activity* berakhir.

7. Activity Diagram *Sell/Jual Kartu*

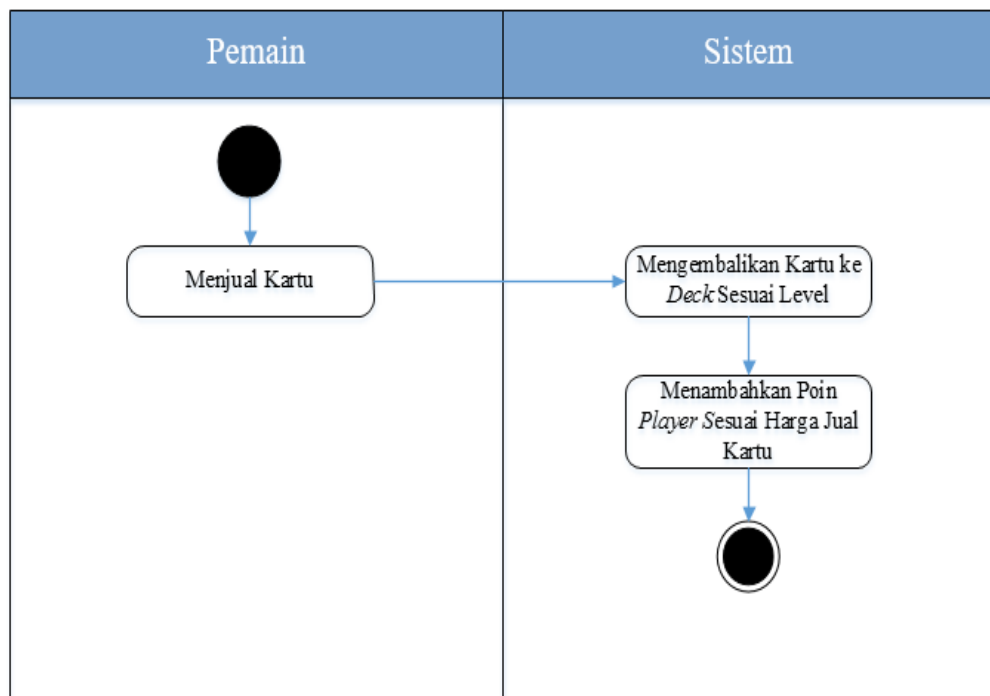
Activity diagram *Sell/Jual Kartu* dapat dilihat pada gambar 4.8, pada gambar tersebut menjelaskan *activity diagram* dimulai dari *player* melakukan *Sell/Jual* kartu. Setelah itu sistem akan mengembalikan kartu pada *deck* kartu sesuai level kartu. Lalu, sistem akan menambahkan poin sesuai harga kartu. Setelah itu *activity* berakhir.



Gambar 4.6 Activity Diagram Strategy phase [8]

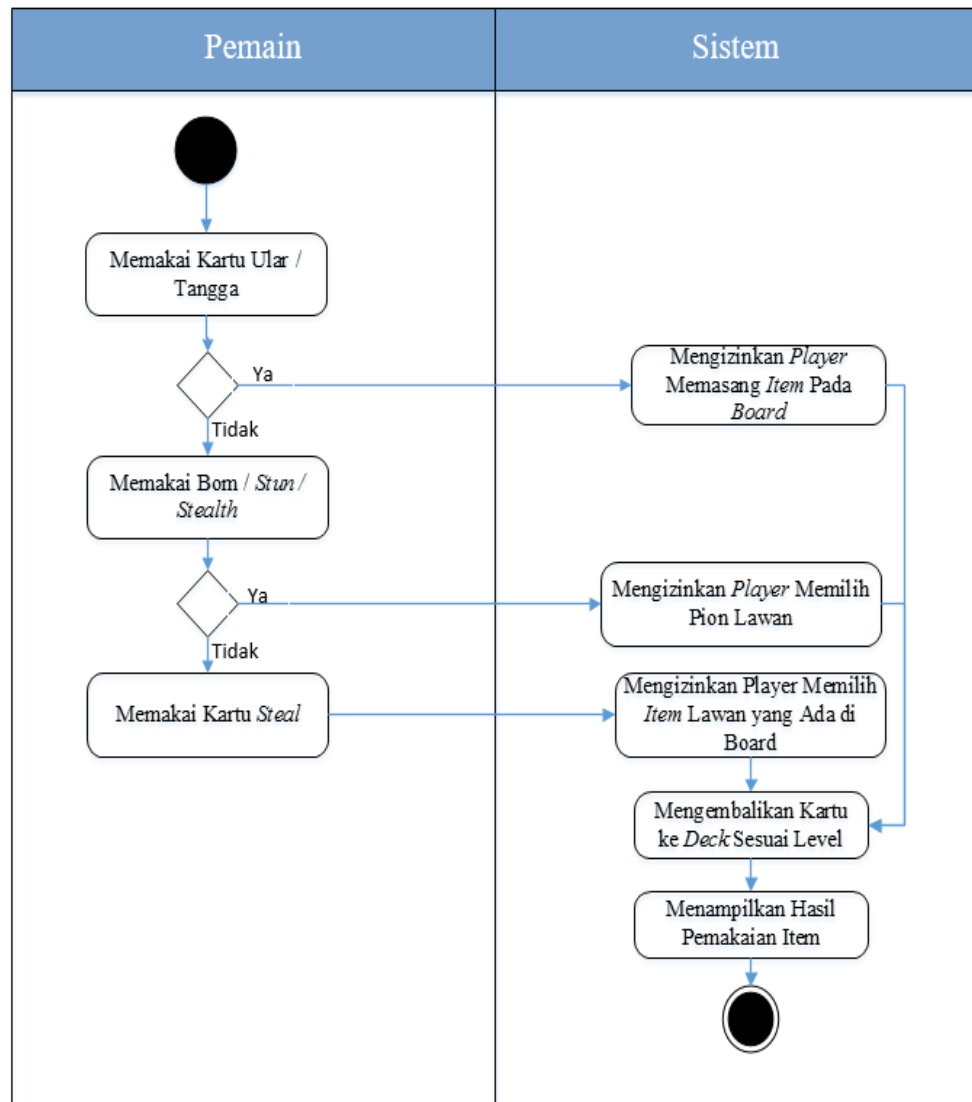


Gambar 4.7 Activity Diagram Buy/Beli Kartu [8]



Gambar 4.8 Activity Diagram Sell/Jual Kartu [8]

8. Activity Diagram Use/Pakai Kartu



Gambar 4.9 Activity Diagram Use/Pakai Kartu [8]

Gambar 4.9 merupakan gambar *activity* diagram Use/Pakai Kartu, Gambar tersebut menjelaskan *Activity* dimulai dari ketika *player* melakukan pemakaian kartu, ada tiga hal yang dapat terjadi. Pertama, *player* memakai kartu ular atau tangga. Ketika *player* “Ya” memakai kartu tersebut maka sistem akan mengizinkan *player* untuk memasang item pada board. Lalu kartu tersebut akan dikembalikan pada *deck* kartu sesuai level kartu. Setelah itu sistem akan menampilkan hasil pemakaian *item*.

Kedua, *player* memakai kartu *bom*, *stun* atau *stealth*. Ketika *player* “Ya” memakai kartu tersebut maka sistem akan mengizinkan *player* untuk memilih pion milik lawan untuk dikenakan *item* yang digunakan. Lalu kartu tersebut akan dikembalikan pada *deck* kartu sesuai level kartu. Setelah itu sistem akan menampilkan hasil pemakaian *item*.

Ketiga, *player* memakai kartu *steal*. Ketika *player* “Ya” memakai kartu tersebut maka sistem akan mengizinkan *player* untuk memilih item milik lawan yang ada pada *board*. Lalu kartu tersebut akan dikembalikan pada *deck* kartu sesuai level kartu. Setelah itu sistem akan menampilkan hasil pemakaian kartu.

4.2 Production

Tahap *Production* adalah tahap pengimplementasian dari *Pre-Production* yang telah dibuat. Tahap ini dilakukan setelah tahap *Pre-Production* selesai. Dalam tahap ini dilakukan proses *Production Scoping*, perancangan sistem yang dibutuhkan, menspesifikasi hardware dan software yang diperlukan oleh pengguna, dan melakukan implementasi dari sistem yang sudah dirancang.

4.2.1 Production Scoping

Dalam membangun sebuah permainan, pada prosesnya selalu bertahap. Oleh karena itu dibutuhkan *Production Scoping* pada penelitian ini untuk memilih bagian-bagian yang perlu dibuat pada tiap tahap pengembangannya. Karena pada penelitian ini adalah tahap pengembangan yang pertama, tidak semua hal yang ada di GDD pada penelitian sebelumnya perlu untuk diimplementasikan.

a. Minimum Elemen

Dari GDD pada penelitian sebelumnya, ada beberapa hal yang harus ada pada permainan ular tangga bergenre strategi ini, karena jika hal-hal ini tidak ada permainan tidak akan berjalan. Hal-hal yang pasti ada dalam permainan yang akan dibangun adalah sebagai berikut:

1. *Starting player*
2. *Starting game*

3. *Turn phase*
4. Kotak start
5. *Board*
6. Dadu
7. Kotak Finish
8. Pion
9. Status area

Untuk hal-hal lainnya dijelaskan pada poin selanjutnya.

b. *Strategy Phase*

Dalam *rule strategy phase* ada 5 hal yang bisa dilakukan oleh pemain yaitu *buy/Beli*, *use/pakai*, *save/simpan*, *sell/jual*, dan *end*. Dalam pembangunan permainan ular tangga bergenre strategi ini pemain hanya bisa melakukan 4 hal saja yaitu *buy/Beli*, *use/pakai*, *sell/jual*, dan *end*. Karena menurut penulis dengan 4 hal tersebut untuk tahap pengembangan pertama permainan sudah bisa berjalan dan bisa dimainkan.

c. Poin

Dalam penelitian ini untuk poin akan dibuat sesuai dengan aturannya yang ada di GDD pada penelitian sebelumnya.

d. Poin *Bar*

Dalam penelitian ini untuk poin bar juga akan dibuat untuk menyimpan poin milik setiap pemain. Poin bar memiliki batas maksimal untuk menyimpan poin yaitu 20 seperti pada GDD penelitian sebelumnya.

e. *Card Shop*

Dalam penelitian ini *card shop* juga akan dibuat dengan isi dan ketentuan seperti yang di GDD pada penelitian sebelumnya.

f. *Save Area*

Karena dalam *strategy phase* pemain tidak bisa *save/simpan* kartu, maka *save area* tidak akan dibuat pada penelitian ini.

g. *Card On Hand*

Dalam penelitian ini *card on hand* akan dibuat karena untuk mengetahui kartu yang didapat oleh pemain ketika membeli kartu.

h. Kartu

Dalam penelitian ini, kartu yang dibuat hanya kartu ular dan tangga biasa baik di level 1, level 2, dan level 3. Karena tahap ini adalah tahap pengembangan yang pertama, penulis ingin memfokuskan pada permainan untuk bisa dimainkan terlebih dahulu. jadi dengan kartu tangga sama ular biasa sudah cukup untuk tahap pengembangan pertama permainan.

i. *Item*

Karena kartu yang dibuat hanya kartu ular dan tangga biasa, maka *item* yang dibuat juga hanya ular dan tangga biasa baik di level 1, level 2, dan level 3 untuk menyesuaikan dengan kartunya.

j. Daftar Kebutuhan

Berdasarkan *scoping* yang telah dijelaskan, fitur-fitur yang akan dibuat dalam permainan ular tangga bergenre strategi ini adalah:

1. *Starting player* adalah fitur yang digunakan untuk menentukan urutan main *player*/pemain.
2. *Starting Game*, adalah fitur untuk mempersiapkan kondisi *game* sebelum *game* dimulai. hal yang perlu disiapkan adalah posisi pion milik setiap *player* berada di kotak start, Keadaan *board* kosong, tanpa ada *item* didalamnya, kondisi poin bar pada setiap *player* dalam keadaan kosong tanpa poin, kondisi kartu pada tiap level dalam posisi teracak, dan *player* akan bermain secara berurutan sesuai dengan urutan yang didapat dari *Starting Player*.
3. *Turn phase* adalah fitur untuk pemain menjalankan pion berdasarkan hasil dadu yang didapat.
4. *Strategy phase* adalah fitur diaman pemain melakukan strategi yang akan dilakukan. didalam *strategy phase* ada 4 fitur yaitu:

- a) *buy*/beli adalah fitur di mana pemain dapat membeli kartu level 1, level 2, atau level 3.
 - b) *use*/pakai adalah fitur di mana pemain menggunakan kartu yang didapat dari membeli.
 - c) *sell*/jual adalah fitur di mana pemain menjual kartu seharga jual kartu yang sudah ditentukan yang didapatkan dari membeli.
 - d) *End* adalah fitur di mana pemain mengakhiri gilirannya dan bergantian dengan pemain lain.
5. Kotak *start* adalah tempat yang digunakan sebagai area pion untuk memulai permainan.
 6. Board merupakan arena yang digunakan untuk memainkan permainannya. Board berisikan 100 kotak yang memiliki angka dari 1-100.
 7. Dadu merupakan alat yang digunakan untuk menentukan jumlah langkah pemain dan jumlah poin yang didapatkan.
 8. Kotak *Finish* merupakan kotak yang dituju untuk memenangkan permainan. Kotak finish berada di kotak 100.
 9. Pion merupakan alat yang digunakan sebagai karakter pemain dalam menjalankan permainan.
 10. Status area merupakan area yang menunjukkan status pemain yang sedang bermain.
 11. Poin merupakan alat yang digunakan pemain untuk membeli kartu.
 12. Poin Bar merupakan sebuah wadah yang digunakan untuk mengumpulkan jumlah poin milik pemain.
 13. *Card Shop* adalah sebuah area yang menyediakan kartu yang dapat dibeli oleh pemain. berisi kartu Level 1, kartu Level 2 dan kartu Level 3.
 14. *Card on Hand* adalah area yang menunjukkan kartu yang didapat pemain.
 15. Kartu adalah fitur untuk pemain mendapatkan item ular atau tangga. Kartu memiliki 3 level, diantaranya:
 - a) Level 1, berisikan kartu yang totalnya 24 yang rincianya adalah 12 kartu ular biasa dan 12 kartu tangga biasa.

- b) Level 2, berisikan kartu yang totalnya 16 yang rinciannya adalah 8 kartu ular biasa dan 8 kartu tangga biasa.
 - c) Level 3, berisikan kartu yang totalnya 18 yang rinciannya adalah 9 kartu ular biasa dan 9 kartu tangga biasa.
16. *Item* merupakan alat/benda yang digunakan pemain sebagai alat bantu untuk menjalankan strategi selama permainan berlangsung yang didapat dari kartu yang dibeli. *Item* bisa dipasang/digunakan bebas pada papan permainan selama memenuhi aturannya. *item* memiliki 3 level sama seperti kartu, diantaranya:
- a) Level 1, berisikan *item* yang totalnya 24 yang rinciannya adalah 12 kartu ular biasa dan 12 kartu tangga biasa.
 - b) Level 2, berisikan *item* yang totalnya 16 yang rinciannya adalah 8 kartu ular biasa dan 8 kartu tangga biasa.
 - c) Level 3, berisikan *item* yang totalnya 18 yang rinciannya adalah 9 kartu ular biasa dan 9 kartu tangga biasa.

4.2.2 Perancangan Sistem

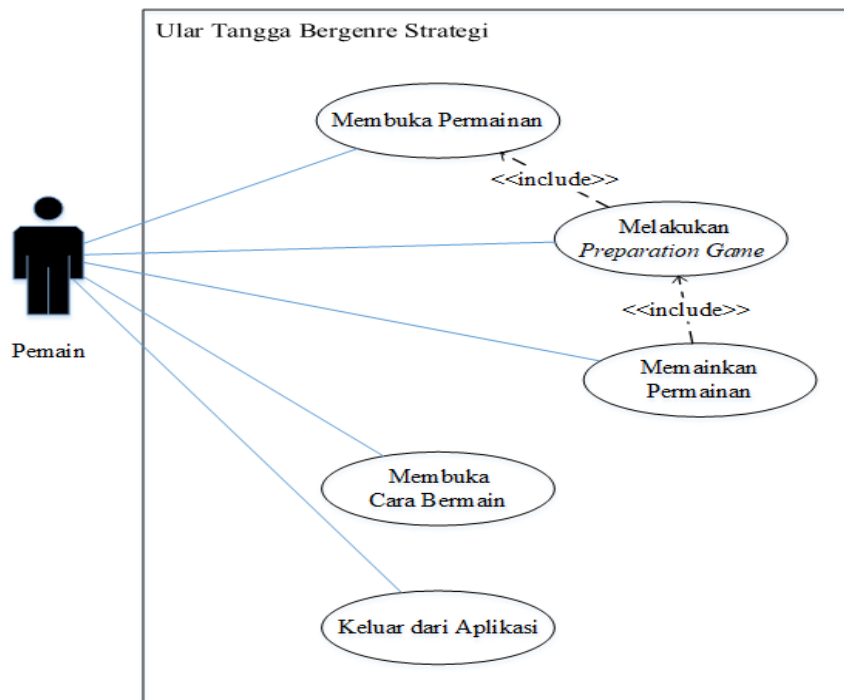
a. *Usecase* dan *Activity* Diagram Baru

Dikarenakan ada proses *Production Scoping* maka diperlukan *Software Requirements Specification*(SRS) yang sesuai dengan permainan yang akan dibangun. Oleh karena itu, penulis membuat SRS baru yang menyesuaikan untuk permainan yang akan dibuat. SRS yang dihasilkan adalah berupa *usecase* diagram dan *activity* diagram yang sesuai dengan permainan yang akan dibangun. Untuk *usecase* dan *activity* diagram yang dihasilkan adalah sebagai berikut:

1. Use Case Diagram

Gambar 4.10 merupakan gambar dari *use case* diagram sistem permainan ular tangga bergenre strategi yang baru. Pada gambar tersebut dijelaskan yang berperan sebagai aktor adalah pemain. Ada 5 *case* pada sistem permainan ular tangga bergenre strategi ini. *Case* pertama adalah membuka permainan untuk memainkan permainannya. *Case* kedua adalah

melakukan *preparation game*, *case ini* digunakan pemain untuk menentukan urutan main *Player/pemain*, sebelum melakukan *preparation game* ada *include* untuk membuka permainan terlebih dahulu, jadi *player* tidak bisa melakukan *preparation game* sebelum menjalankan *case* membuka permainan. *Case* ketiga adalah memainkan permainan, pada bagian ini pemain akan memainkan permainannya namun sebelum pemain memainkan permainan ada *include* juga yang berarti melakukan *starting player* terlebih dahulu untuk memainkan permainannya. *Case* yang keempat adalah membuka cara bermain digunakan untuk melihat tutorial permainan. Dan *case* yang kelima adalah Keluar aplikasi.



Gambar 4.10 Usecase diagram sistem permainan ular tangga bergenre strategi yang baru

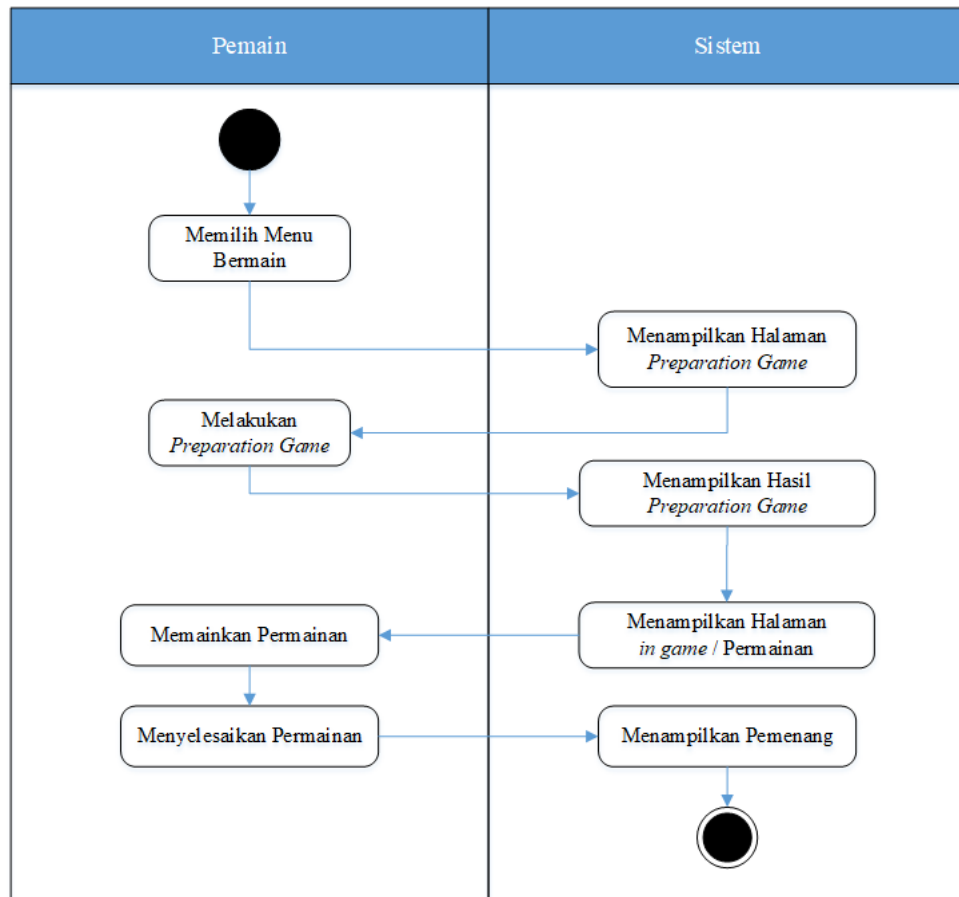
2. Activity Diagram

Ada 10 *activity diagram* dalam permainan ular tangga bergenre strategi ini. berikut ini adalah 10 *activity diagram* tersebut.

a) Activity Membuka Permainan

Activity diagram membuka permainan dapat dilihat pada gambar 4.11, pada gambar tersebut menjelaskan *activity* dimulai dari pemain memilih menu bermain. Setelah pemain memilih menu bermain, sistem

akan menampilkan halaman *Preparation Game*. Pada halaman *preparation game* pemain melakukan *preparation game*. Untuk penjelasan *preparation game* dijelaskan pada *activity preparation game*. Setelah *preparation game* selesai, pemain akan memainkan permainan hingga salah satu pemain memenangkan permainannya.

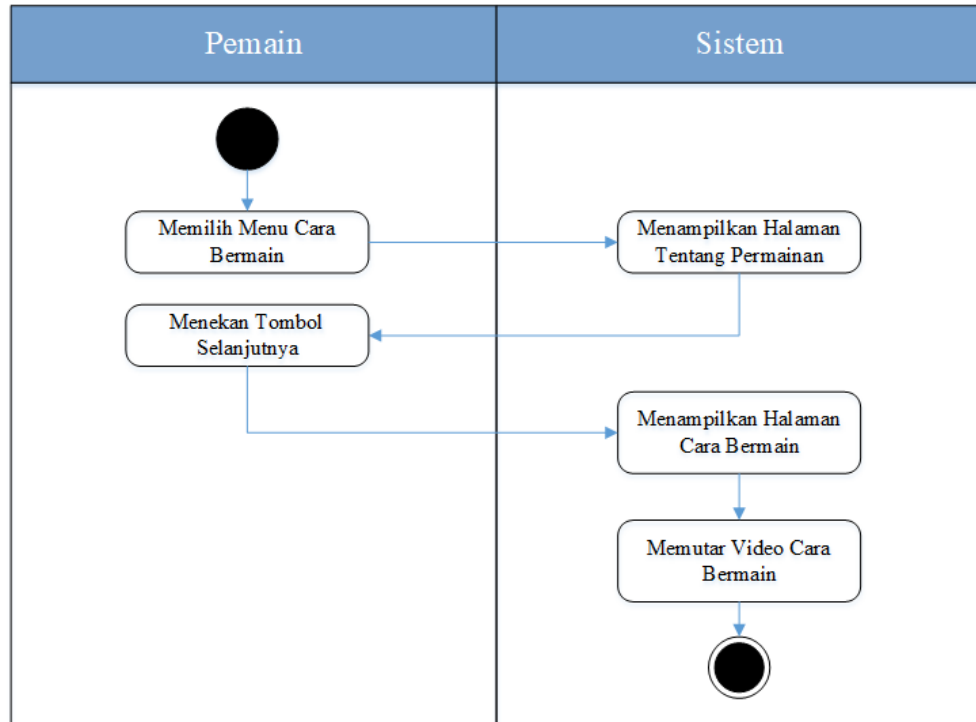


Gambar 4.11 Activity Diagram Membuka Permainan

b) Activity Membuka Cara Bermain

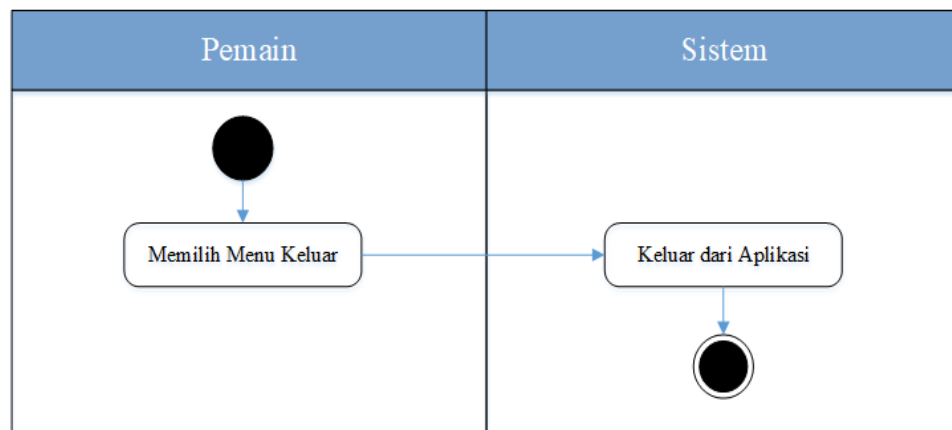
Activity diagram membuka cara bermain dapat dilihat pada gambar 4.12, pada gambar tersebut menjelaskan *activity* dimulai dari pemain memilih menu cara bermain. Setelah pemain memilih menu cara bermain, sistem akan menampilkan halaman tentang permainan. Halaman ini berisi tentang penjelasan permainan ular tangga bergenre strategi. Pada halaman tentang permainan, pemain menekan tombol selanjutnya untuk menampilkan halaman cara bermain. Halaman yang berisi video tutorial

memainkan permainan ini. Setelah halaman cara Bermain ditampilkan, sistem akan langsung memutar video cara bermain.



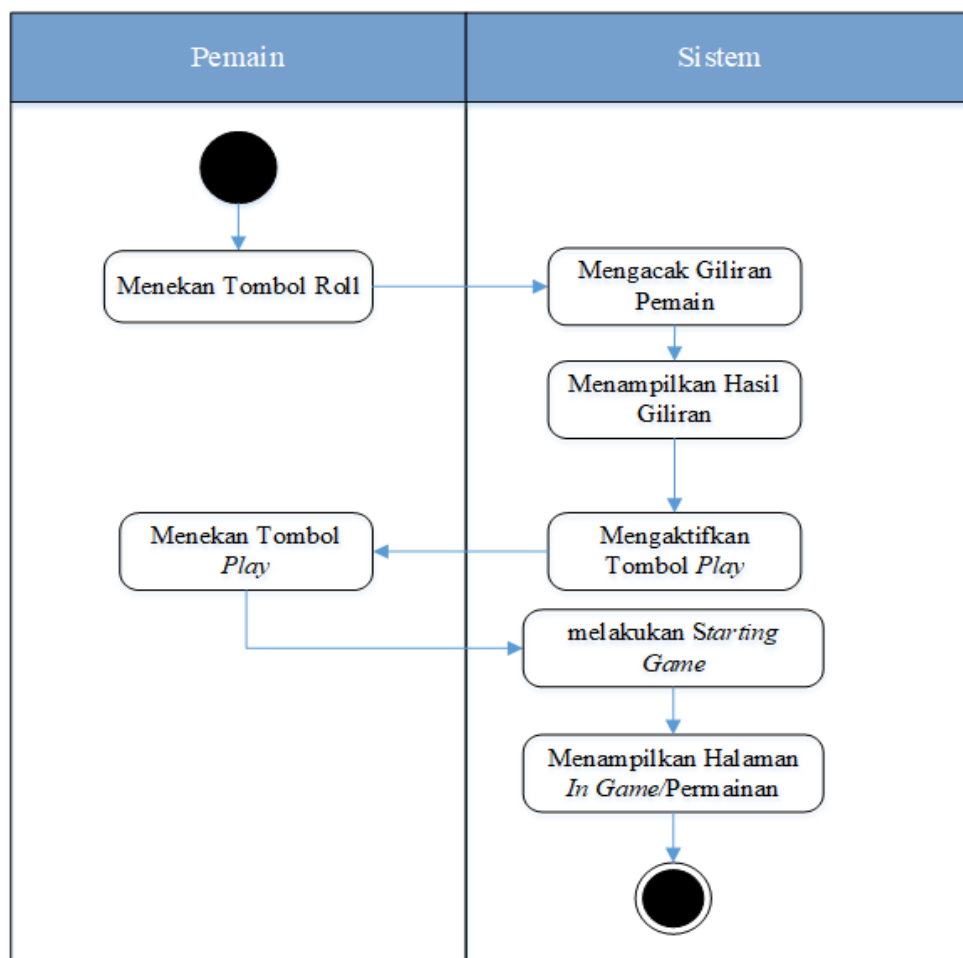
Gambar 4.12 Activity Diagram Menu Cara Bermain

c) Activity Keluar dari Aplikasi



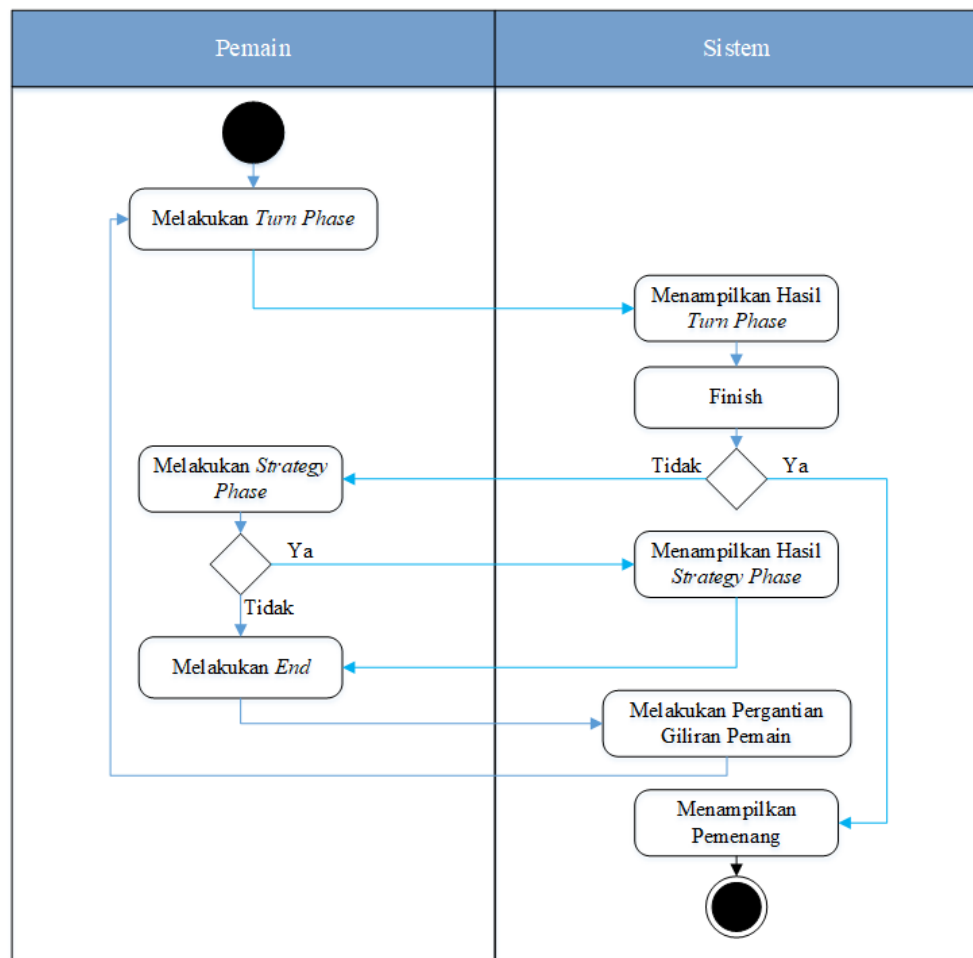
Gambar 4.13 Activity Diagram Menu Keluar

Gambar 4.13 merupakan gambar *activity* keluar dari aplikasi. Gambar tersebut menjelaskan *activity* dimulai dari pemain memilih menu Keluar. Setelah pemain memilih menu Keluar, sistem akan langsung menutup aplikasi.

d) *Activity Preparation Game*

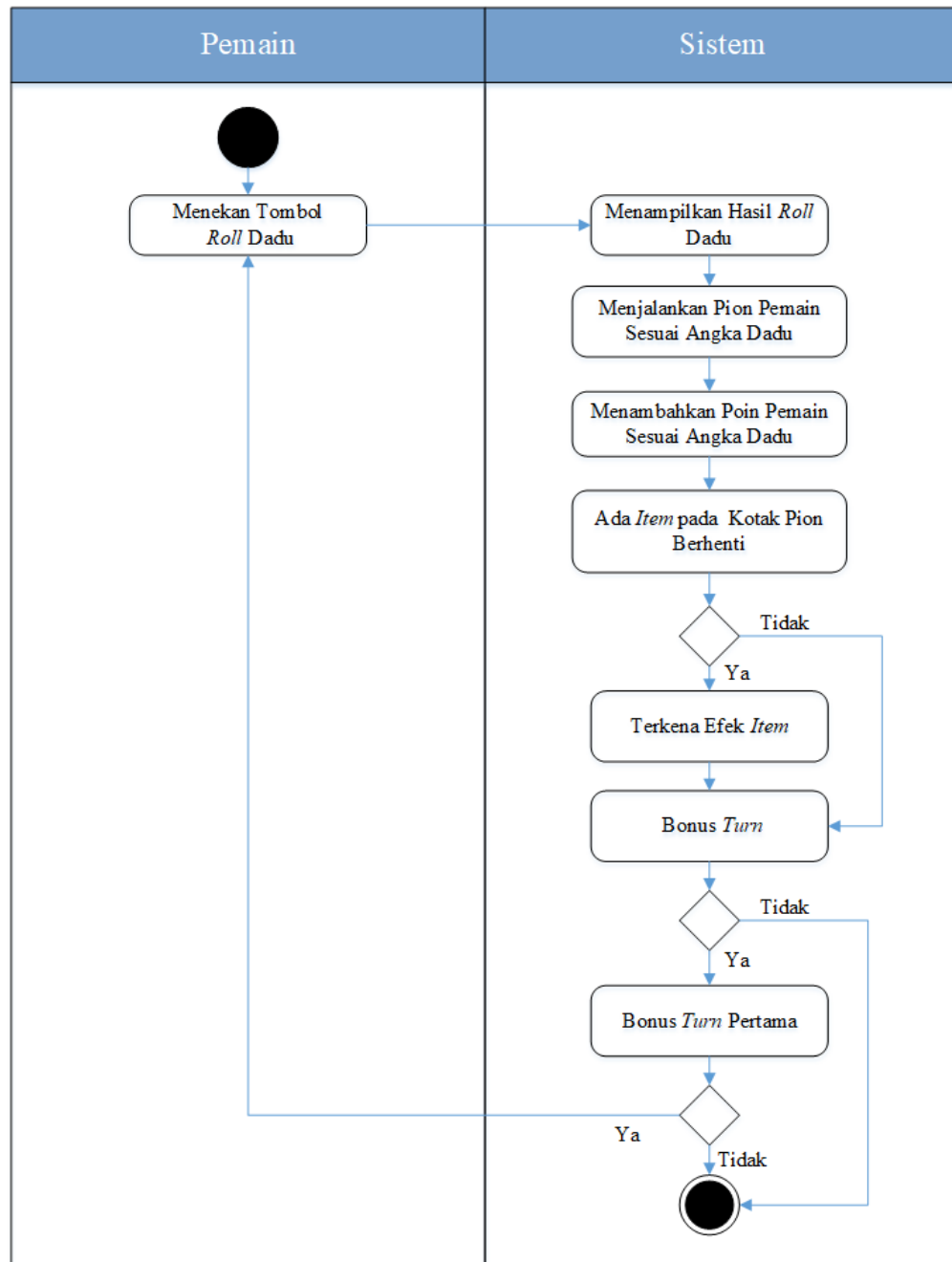
Gambar 4.14 Activity Diagram Preparation Game

Gambar 4.14 merupakan gambar *activity diagram Preparation Game*, gambar tersebut menjelaskan *activity* dimulai dari salah satu pemain menekan tombol Roll untuk menentukan urutan main. setelah itu, sistem akan mengacak giliran pemain. Setelah teracak, sistem memunculkan hasil giliran pemain. Hasil tersebut berisi angka giliran yang didapat. Setelah hasil dari giliran pemain muncul, sistem mengaktifkan tombol *Play* untuk menuju ke permainannya. Ketika tombol *Play* ditekan pada saat aktif, sistem akan melakukan *starting game* yaitu mempersiapkan permainannya. Hal-hal yang harus dipersiapkan sistem dapat dilihat pada bab 4 poin 4.2.1. Setelah *starting game* selesai dilakukan, sistem menampilkan halaman *in game/permainan*.

e) *Activity Permainan*

Gambar 4.15 Activity Diagram Permainan

Gambar 4.15 merupakan gambar *activity* diagram permainan, Gambar tersebut menjelaskan *activity* dimulai dari *player* melakukan *Turn phase*. Pada bagian *Turn Phase* dijelaskan pada *activity Turn Phase*. Setelah *Turn phase* dilakukan, sistem akan memeriksa apakah pion sudah sampai finish, Jika “Ya” maka permainan berakhir dan sistem akan menampilkan pemenang dari permainan. Jika “Tidak”, maka pemain akan melakukan *Strategy phase*, Pada bagian *Strategy Phase*, ada 4 tindakan yang dapat dilakukan oleh pemain yaitu beli kartu, pakai kartu, jual kartu, dan *End*. *End* merupakan tindakan untuk mengakhiri *Strategy Phase*. Setelah *player* melakukan *End*, sistem akan melakukan pergantian *player* sesuai urutan main dan akan memulai *Turn phase* lagi dari awal.

f) *Activity Turn phase*

Gambar 4.16 Activity Diagram Turn Phase

Gambar 4.16 merupakan gambar *activity Diagram turn phase*, pada gambar tersebut menjelaskan *activity diagram* dimulai dari pemain Tombol *roll* dadu untuk menentukan jumlah langkah yang didapat pemain.

Selanjutnya sistem akan menjalankan pion dan menambah poin milik pemain sesuai angka jumlah langkah yang didapat pemain.

Setelah pion berjalan, sistem akan melakukan pemeriksaan pada kotak tempat pion berhenti ada *item* atau tidak. Jika “Tidak” sistem akan langsung memeriksa bonus *turn*. Namun jika “Ya” dan kotak tersebut berisi kepala ular milik lawan, maka pion akan turun menuju kotak yang berisi ekor ular. Jika “Ya” dan kotak tersebut berisi kepala ular milik sendiri, maka pion tidak akan turun. Jika “Ya” dan kotak berisi kaki tangga sendiri pion akan naik menuju kepala/ujung tangga. Jika “Ya” dan kotak berisi kaki tangga milik lawan maka pion tidak akan naik.

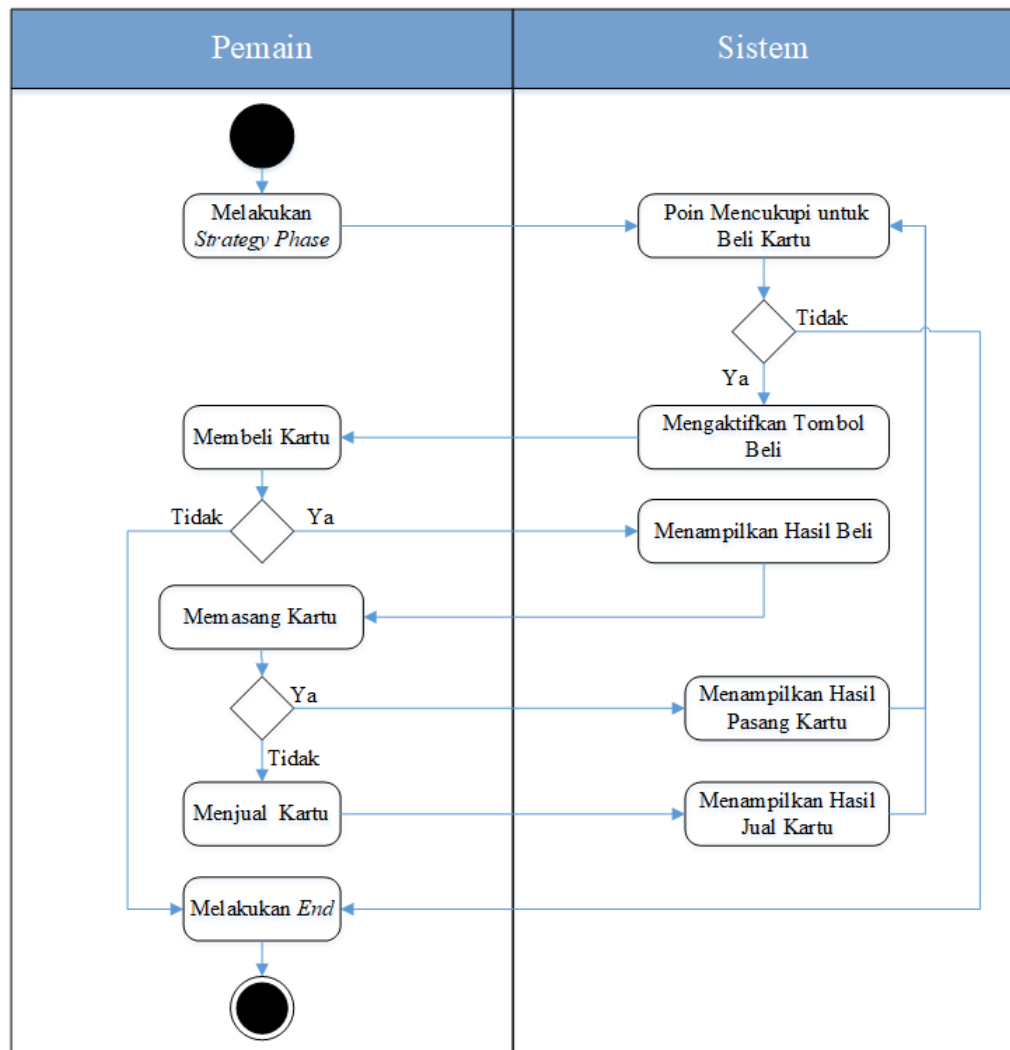
Setelah pemeriksaan pada kotak tempat pion berhenti selesai dan menjalankan efek seperti yang ditentukan, sistem akan memeriksa bonus *turn*. Bonus *turn* adalah ketika angka roll dadu pemain mendapat 6. Jika pemain tidak mendapat bonus *turn* maka *Turn phase* langsung berakhir. Tetapi jika *player* mendapat bonus *turn*, sistem akan melakukan pemeriksaan kondisi apakah bonus *turn* yang didapat merupakan bonus *turn* yang pertama, jika “Ya” maka *player* dapat melakukan roll dadu kembali. Jika “Tidak” maka *Turn phase* berakhir.

g) *Activity Strategy phase*

Activity diagram *strategy phase* dapat dilihat pada gambar 4.17, pada gambar tersebut menjelaskan *activity* dimulai pemain melakukan strategi. Setelah itu sistem memeriksa poin apakah mencukupi atau tidak, ketika poin pemain tidak mencukupi maka pemain harus langsung melakukan end. Namun ketika poin mencukupi untuk membeli kartu pemain diberi pilihan membeli kartu atau tidak, ketika pemain tidak membeli kartu maka pemain harus melakukan end. Tetapi, ketika pemain membeli kartu, sistem akan menampilkan hasil beli.

Setelah membeli kartu pemain akan diberi pilihan lagi, memasang kartu yang dibeli atau tidak. Ketika pemain memilih pasang, sistem akan menampilkan hasil pasang dan setelah hasil pasang tampil sistem akan kembali memeriksa jumlah poin. Namun jika pemain tidak memasang kartu,

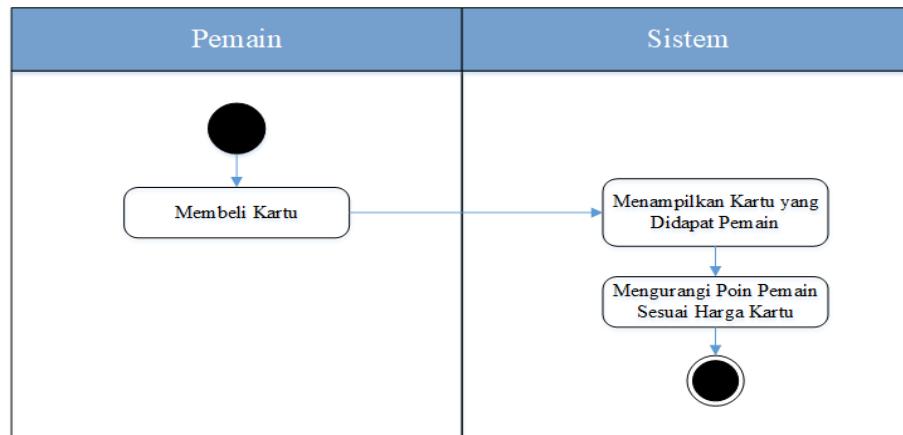
maka pemain akan menjual kartu yang didapat. Setelah itu sistem akan menampilkan hasil jual dan kemudian sistem akan kembali memeriksa jumlah poin.



Gambar 4.17 Activity Diagram Strategy Phase

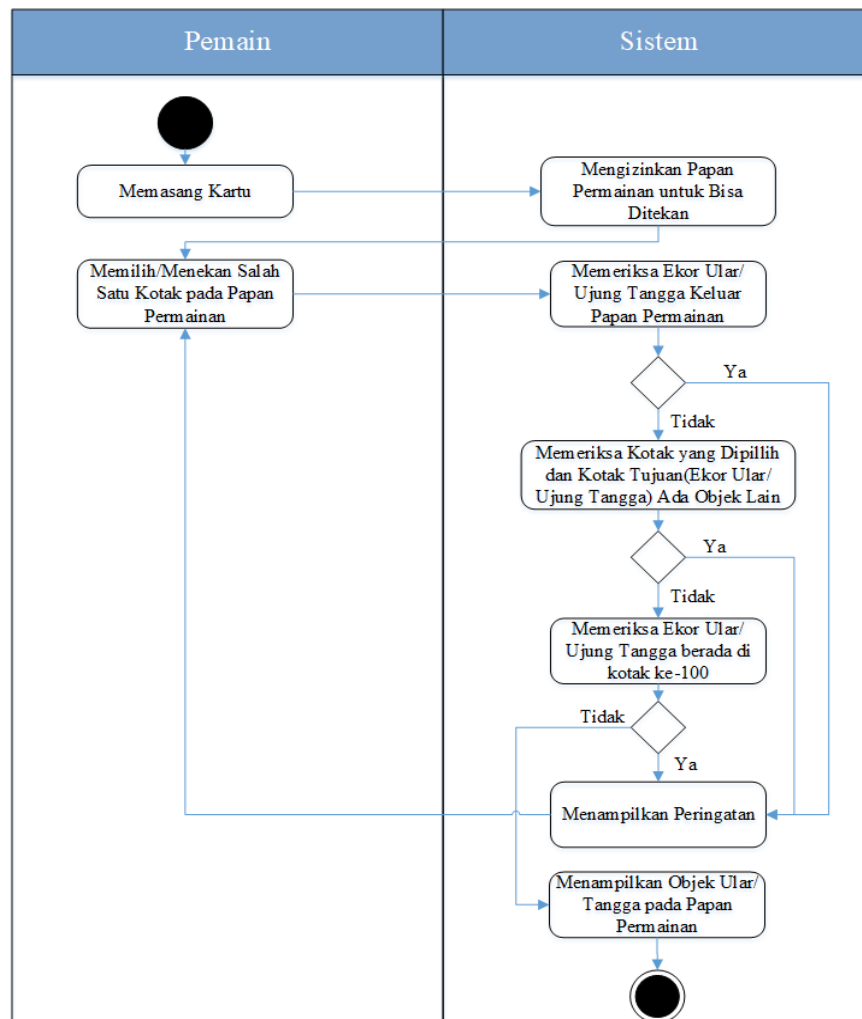
h) Activity Beli Kartu

Activity diagram Beli Kartu dapat dilihat pada gambar 4.18, pada gambar tersebut menjelaskan activity diagram dimulai dari pemain melakukan beli kartu. Setelah kartu terbeli, sistem akan menampilkan kartu yang didapatkan. Lalu, sistem akan mengurangi poin milik pemain sesuai harga kartu. Setelah itu activity berakhir.



Gambar 4.18 Activity Diagram Beli Kartu

i) Activity Pasang Kartu

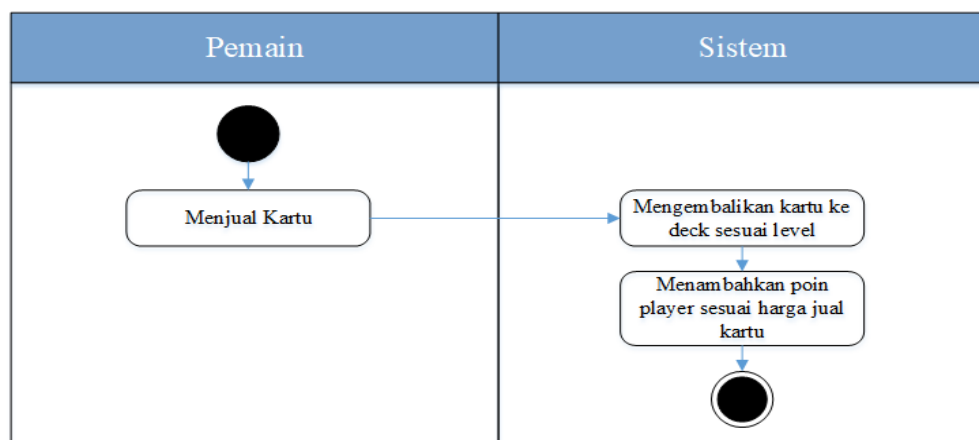


Gambar 4.19 Activity Diagram Pasang Kartu

Gambar 4.19 merupakan gambar *activity* Diagram Pasang Kartu, pada gambar tersebut menjelaskan *activity* dimulai dari pemain memilih pasang kartu. Setelah itu sistem akan membuka papan permainan agar bisa ditekan. Setelah papan terbuka. Pemain memilih kotak pada papan permainan dengan cara menekan(*click*) kotak yang dipilih.

Setelah kotak ditekan, sistem akan memeriksa ekor ular / ujung tangga keluar arena atau tidak, ketika ekor ular / ujung tangga keluar arena permainan, sistem akan menampilkan peringatan, dan pemain akan memilih ulang kotaknya dengan kotak yang lain. Ketika ekor ular / ujung tangga tidak keluar arena permainan, sistem akan memeriksa kotak yang dipilih dan kotak tujuan (ekor ular/ujung tangga) apakah ada objek lain, ketika kotak yang dipilih dan kotak tujuan ada objek lain, sistem akan menampilkan peringatan, dan pemain akan memilih ulang kotaknya dengan kotak yang lain. Ketika kotak yang dipilih dan kotak tujuan tidak ada objek, sistem akan memeriksa ekor ular / ujung tangga apakah berada di kotak ke-100 atau tidak. Ketika ekor ular / ujung tangga berada di kotak ke-100, sistem akan menampilkan peringatan, dan pemain akan memilih ulang kotaknya dengan kotak yang lain. Ketika ekor ular / ujung tangga tidak berada di kotak ke-100, sistem akan menampilkan objek ular / tangga pada papan permainan sesuai kartu yang didapat.

j) *Activity* Jual Kartu



Gambar 4.20 *Activity* Diagram Jual Kartu

Gambar 4.20 merupakan gambar *activity* diagram jual kartu *Activity* dimulai dari pemain melakukan jual kartu. Setelah itu sistem akan mengembalikan kartu pada deck kartu sesuai levelnya. Lalu, sistem akan menambahkan poin sesuai harga kartu.

b. Perancangan Algoritma

Perancangan algoritma merupakan perancangan yang berfungsi untuk mendeskripsikan algoritma-algoritma yang berada di dalam aplikasi. algoritma dapat dipanggil dengan menyertakan variabel, baik hanya satu variabel, banyak variabel, atau bahkan tidak ada sama sekali. Adapun algoritma-algoritma yang terdapat dalam permainan ular tangga bergenre strategi dijelaskan melalui flowchart sebagai berikut:

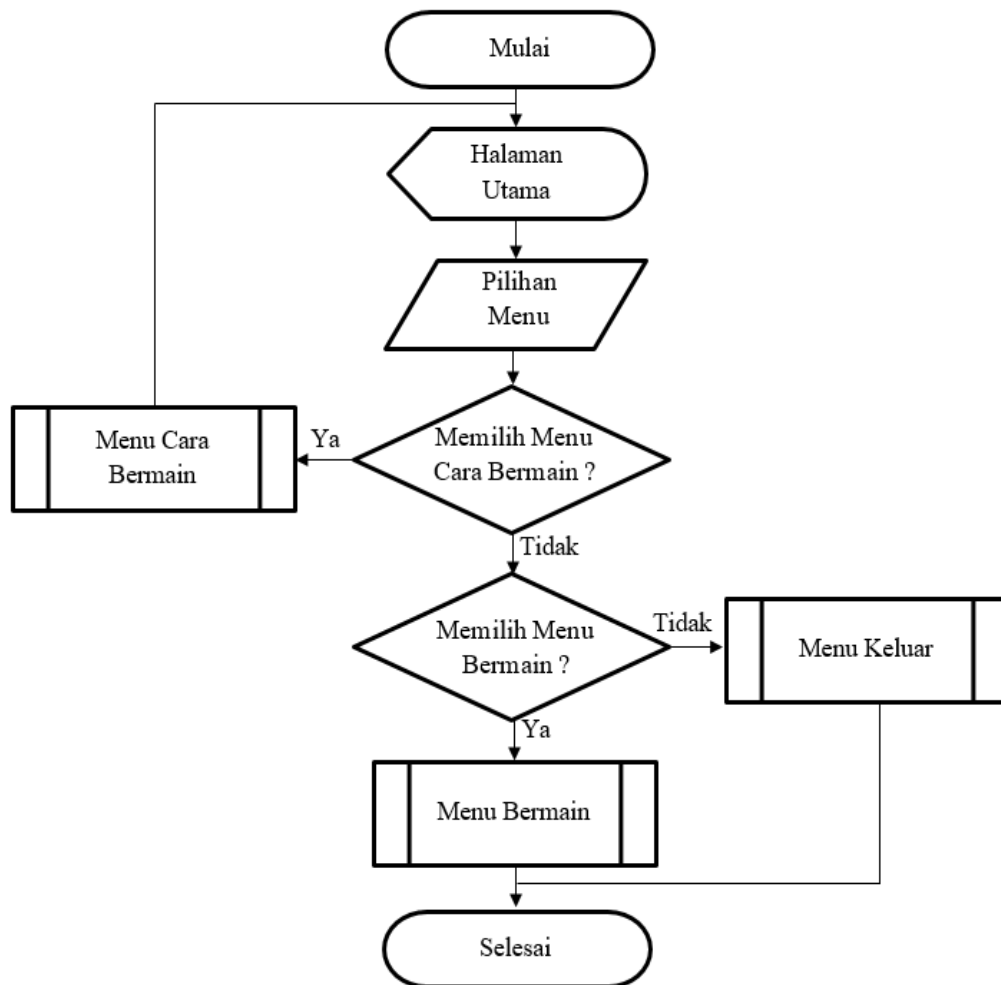
1. Perancangan Algoritma Halaman Utama

Algoritma halaman utama pada permainan ular tangga bergenre strategi dapat dilihat pada Gambar 4.21. pada gambar tersebut dijelaskan permainan secara keseluruhan dari awal sampai akhir, pertama kali ketika aplikasi dijalankan sistem akan menampilkan halaman utama. Pada halaman utama terdapat 3 menu pilihan yaitu bermain, cara bermain, dan keluar.

Menu Bermain adalah menu yang digunakan untuk memasuki permainan untuk penjelasan lebih lengkap dari menu bermain ada pada algoritma menu bermain.

Menu cara bermain adalah menu yang digunakan untuk menampilkan tentang permainan yang dibuat dan tutorial cara memainkannya. Untuk penjelasan lebih lengkap dari menu cara bermain ada pada algoritma menu cara bermain. Ketika semua yang ada pada menu cara bermain sudah dikerjakan sistem akan kembali ke halaman utama.

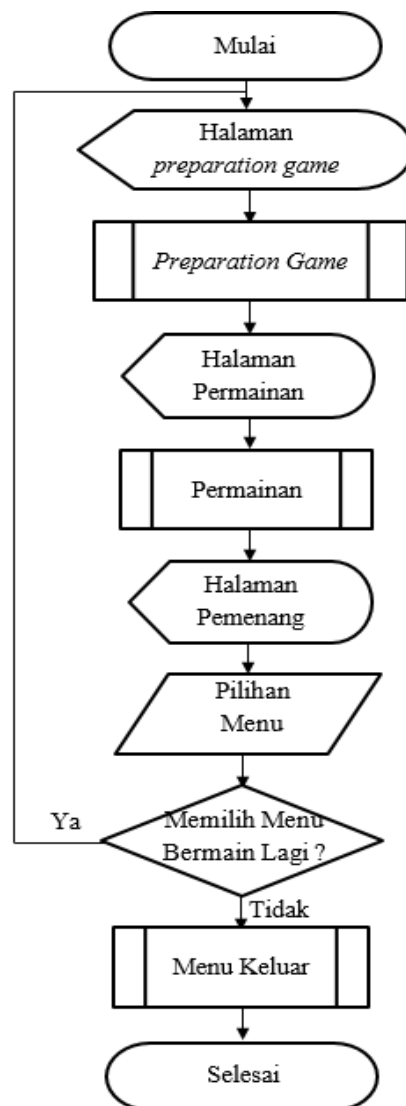
Menu keluar adalah menu yang digunakan untuk keuar dari aplikasi. Untuk penjelasan lebih lengkap dari menu keluar ada pada algoritma menu keluar.



Gambar 4.21 Flowchart Halaman Utama

2. Perancangan Algoritma Menu Bermain

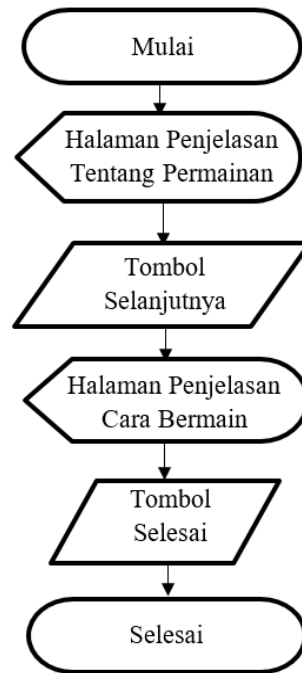
Algoritma Menu Bermain pada permainan ular tangga bergenre strategi dapat dilihat pada gambar 4.22. Pada gambar tersebut dijelaskan ketika pemain memilih menu bermain, sistem akan memunculkan halaman *preparation game*. Untuk penjelasan lebih lengkap dari *preparation game* ada pada algoritma *preparation game*. Setelah *preparation game* selesai sistem akan menampilkan halaman permainan. dan pemain memainkan permainan sampai selesai. Untuk penjelasan lebih lengkap dari permainan ada pada algoritma permainan. Ketika permainan selesai sistem akan menampilkan halaman pemenang, didalam halaman pemenang pemain diberi pilihan bermain lagi atau keluar dari aplikasi.



Gambar 4.22 Flowchart Menu Bermain

3. Perancangan Algoritma Menu Cara Bermain

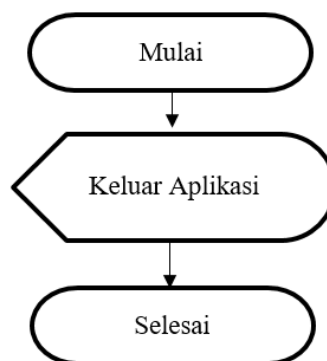
Algoritma Menu Cara Bermain pada permainan ular tangga bergenre strategi dapat dilihat pada gambar 4.23. Pada gambar tersebut dijelaskan ketika pemain memilih menu cara bermain, sistem akan memunculkan halaman penjelasan tentang permainan. Halaman ini berisikan penjelasan tentang permainan ini. Ketika pemain menekan tombol selanjutnya sistem akan menampilkan halaman tutorial cara bermain. Halaman ini berisikan cara-cara memainkan permainan ini. Setelah selesai melihat tutorial, sistem akan mengembalikan tampilan ke halaman utama.



Gambar 4.23 Flowchart Menu Cara Bermain

4. Perancangan Algoritma Menu Keluar

Algoritma Menu Keluar pada permainan ular tangga bergenre strategi dapat dilihat pada gambar 4.24. Pada gambar tersebut dijelaskan ketika pemain memilih menu keluar, sistem akan menampilkan aplikasi tertutup dan keluar.

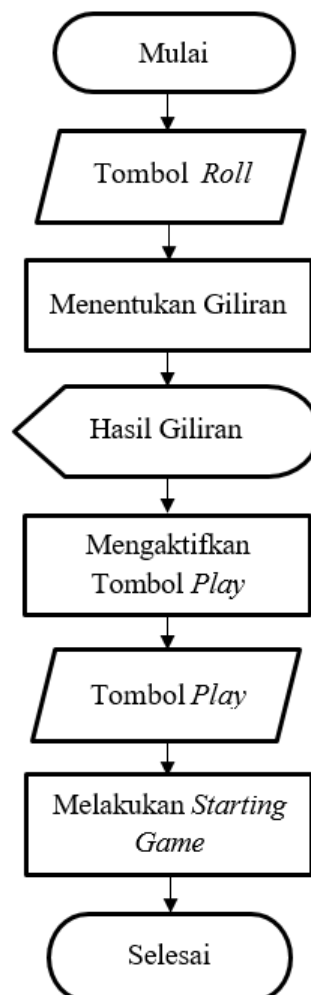


Gambar 4.24 Flowchart Menu Keluar

5. Perancangan Algoritma *Preparation Game*

Algoritma *preparation game* pada permainan ular tangga bergenre strategi dapat dilihat pada gambar 4.25. Pada gambar tersebut dijelaskan, ketika pemain berada pada *preparation game* maka akan masuk ke fase

starting player, yaitu fase untuk menentukan giliran pemain. Fase ini mulai dari player menekan tombol roll, setelah tombol roll ditekan sistem akan memproses dan menentukan giliran tiap pemain, setelah selesai ditentukan, sistem akan menampilkan hasil giliran yang berupa angka yang didapat pemain, angka yang didapat menunjukan giliran pemain, kemudian sistem mengaktifkan tombol play. Setelah tombol play aktif, pemain diharuskan menekan tombol play untuk menuju ke halaman permainan. Setelah tombol play ditekan sistem akan melakukan *starting game*, yaitu mempersiapkan hal-hal pada permainan, setelah *starting game* selesai sistem menampilkan halaman permainan.



Gambar 4.25 Flowchart Perparation Game

6. Perancangan Algoritma Permainan

Algoritma Permainan pada permainan ular tangga bergenre strategi dapat dilihat pada gambar 4.26. Pada gambar tersebut dijelaskan sudah berada didalam permainan. Sistem akan memeriksa giliran sesuai yang didapat pada *preparation game*. Ketika giliran sudah di periksa, pemain yang mendapat giliran akan melakukan *turn phase*. Untuk pejelasan lebih lengkap dari *turn phase* ada pada algoritama *turn phase*.

Setelah *turn phase* selesai, sistem akan memeriksa keberadaan pion, apakah sudah mencapai finish, ketika pion mencapai finish permainan berakhir, tetapi ketika belum mencapai finish pemain diberi 2 pilihan, melakukan *strategy phase* atau tidak. Ketika pemain melakukan starategi, sistem akan menampilkan hasil strategi. Untuk pejelasan lebih lengkap dari *strategy phase* ada pada algoritama *strategy phase*. Setelah selesai melakukan *strategy phase*, pemain menekan tombol selesai. Ketika tidak melakukan *strategy phase*, pemain akan langsung menekan tombol selesai. Tombol selesai adalah tombol untuk mengakhiri giliran.

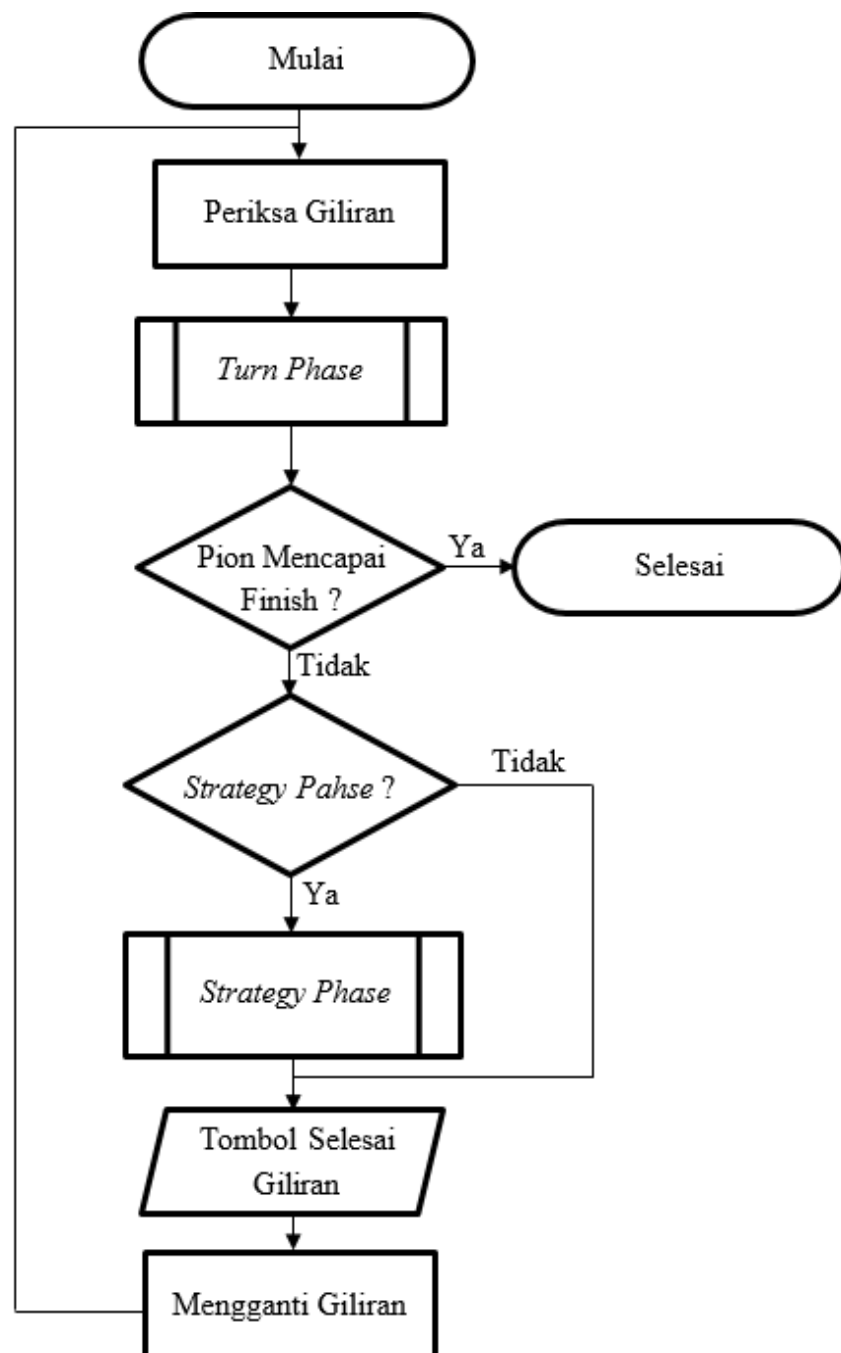
Setelah Tombol selesai ditekan oleh pemain, sistem akan mengganti giliran sesuai dengan urutan. Dan kembali ke sistem memeriksa giliran.

7. Perancangan Algoritma *Turn phase*

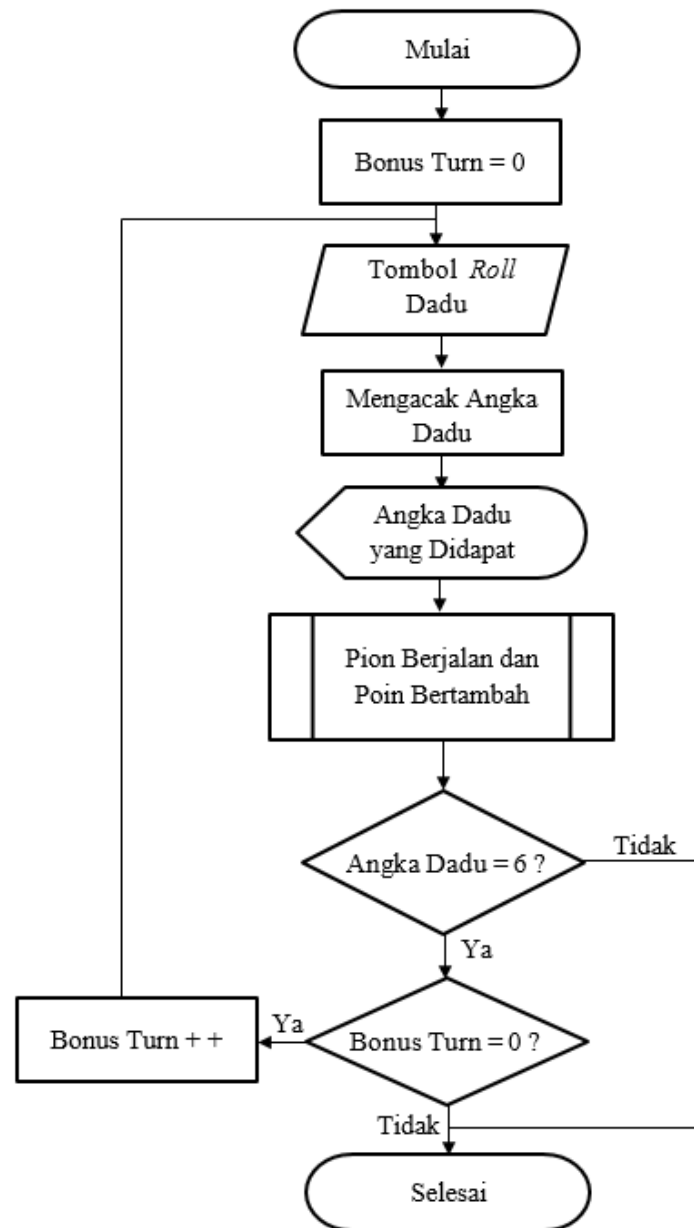
Algoritma *Turn phase* pada permainan ular tangga bergenre strategi dapat dilihat pada gambar 4.27. Pada gambar tersebut dijelaskan *turn phase* mulai dari sistem menjadikan nilai bonus *turn* sama dengan 0. Setelah itu pemain melakukan roll dadu (menekan tombol roll). Setelah roll dadu sistem akan mengacak angka dadu, setelah diacak, sistem akan menampilkan angka yang didapat pemain. kemudian sistem akan menjalankan pion dan menambahkan poin sesuai angka dadu yang didapat. Untuk pejelasan lebih lengkap dari menjalankan pion dan menambahkan poin ada pada algoritama pion berjalan dan poin bertambah.

Ketika angka dadu yang didapat bukan angka 6 *turn phase* akan berakhir, namun ketika angka dadu yang didapat adalah angka 6 sistem

akan memeriksa nilai bonus *turn*, ketika bonus *turn* sama dengan 0 maka sistem akan menambahkan nilai bonus *turn* senilai 1 dan pemain diizinkan roll dadu lagi, tetapi ketika bonus *turn* tidak sama dengan 0 *turn phase* akan berakhir.



Gambar 4.26 Flowchart Permainan

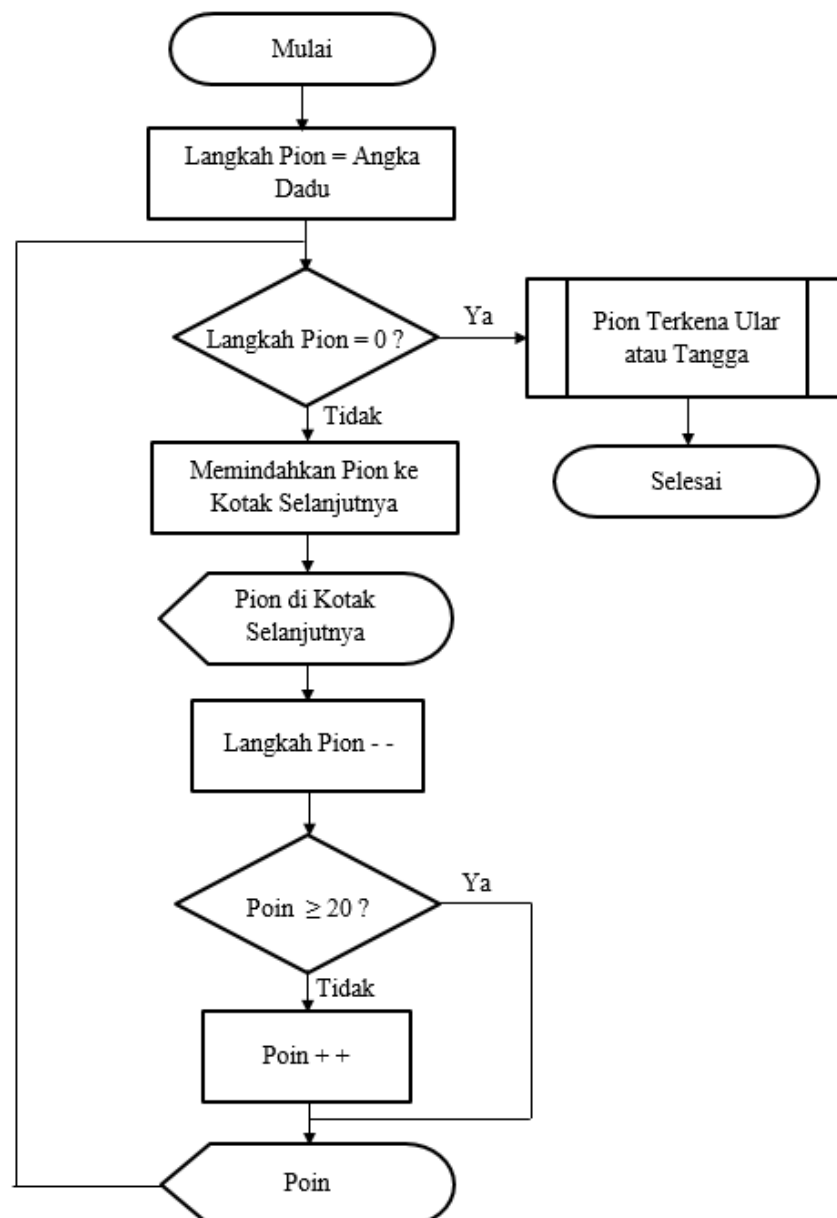


Gambar 4.27 Flowchart *Turn phase*

8. Perancangan Algoritma pion berjalan dan poin bertambah

Algoritma pion berjalan dan poin bertambah pada permainan ular tangga bergenre strategi dapat dilihat pada gambar 4.28. Pada gambar tersebut dijelaskan sistem akan memproses angka dadu yang didapat dijadikan langkah pion. Ketika langkah pion sama dengan 0 maka akan menjalankan algoritma pion terkena ular atau tangga dan algoritma ini akan berakhir, tetapi ketika langkah pion tidak sama dengan 0, sistem akan

memindahkan pion ke kotak selanjutnya, dan sistem akan mengurangi nilai langkah pion sejumlah 1. Setelah itu, sistem akan memeriksa jumlah poin pemain, ketika kurang dari 20 poin pemain akan ditambahkan, namun ketika poin sama dengan 20 maka poin tidak akan ditambahkan. Setelah itu sistem akan memeriksa kembali nilai langkah pion, ketika langkah pion belum = 0 proses ini akan terus berjalan.



Gambar 4.28 Flowchart pion berjalan dan poin bertambah

9. Perancangan algoritma *Strategy phase*

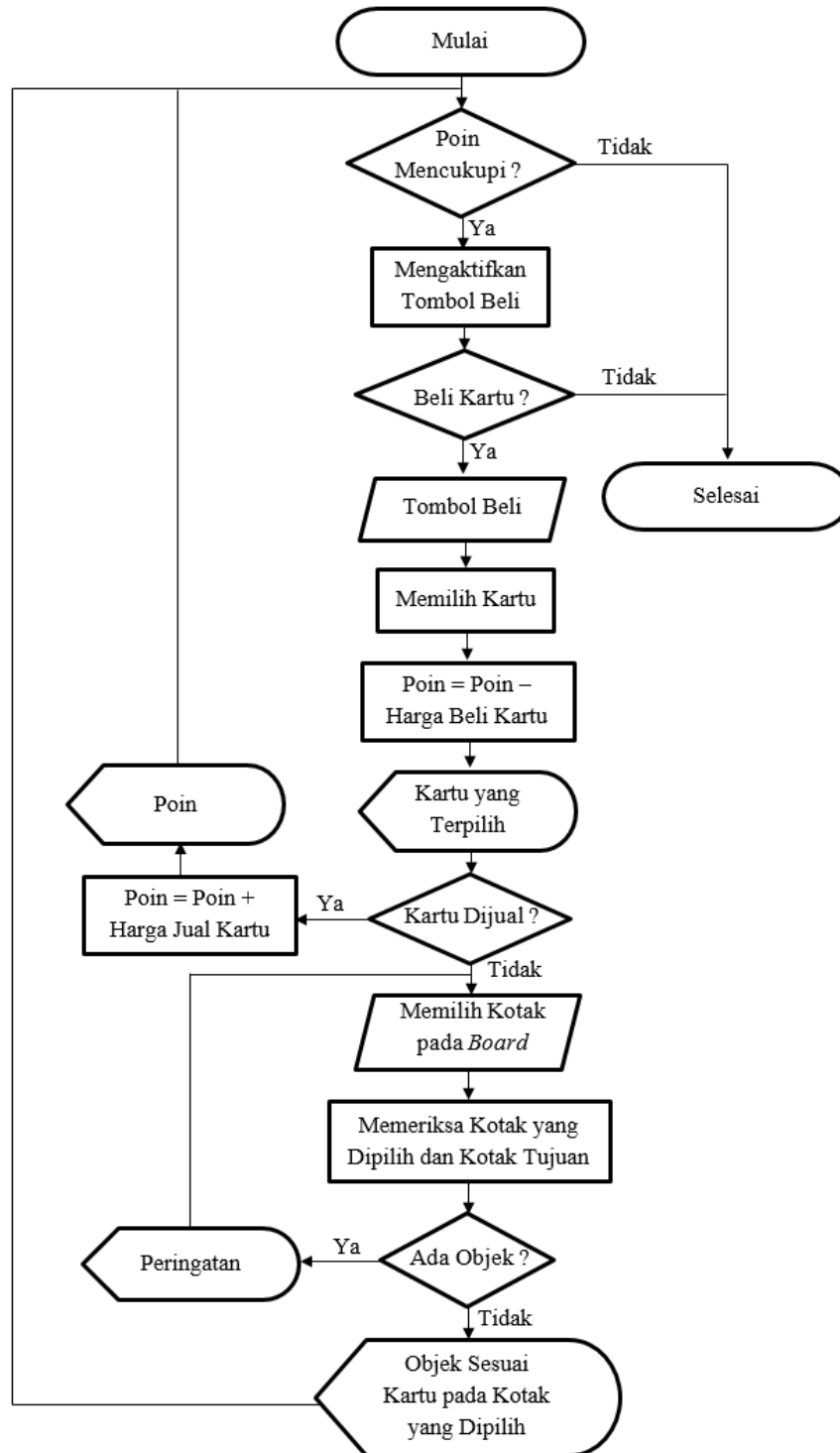
Algoritma *strategy phase* pada permainan ular tangga bergenre strategi dapat dilihat pada gambar 4.29. Pada gambar tersebut dijelaskan pada saat *strategy phase* sistem akan memeriksa poin apakah cukup untuk membeli katu atau tidak, ketika poin tidak mencukupi, maka algoritma akan langsung berakhir, tetapi ketika poin mencukupi sistem akan mengaktifkan tombol beli sesuai jumlah poin. Ketika tombol beli aktif, pemain diberikan 2 pilihan, mau membeli atau tidak, jika tidak membeli maka algoritma akan berakhir, tetapi ketika membeli sistem akan memilihkan kartu dan mengurangi poin sesuai level kartu yang dibeli. Kemudian sistem akan menampilkan kartu yang diperoleh pemain.

Setelah kartu ditampilkan, pemain akan di beri 2 plihkan jual kartu atau pasang kartu, ketika kartu yang didapat dijual maka sistem akan menambahkan poin sesuai harga jual kartu. Dan kartu dikembalikan ke *deck*. Ketika pemain mememilih memasang kartu, sistem akan mengizinkan pemain memilih kotak utuk dipasang *item* sesuai kartu yang didapat. Setelah kotak sudah dipilih, sistem akan memeriksa kotak yang dipilih sama kotak tujuan *item* ada objek atau tidak dan memeriksa apakah ujung tangga/ekor ular keluar arena atau tidak, ketika ada objek atau keluar arena sistem akan memunculkan peringatan dan pemain disuruh memilih ulang. Namun ketika kotak yang dipilih dan tujuan kotak tidak ada objek, dan tidak keluar arena, sistem akan memunculkan objek sesuai dari kartu yang didapat dan kartu dikembalikan ke *deck* dan sistem akan kembali memeriksa poin cukup atau tidak untuk membeli kartu.

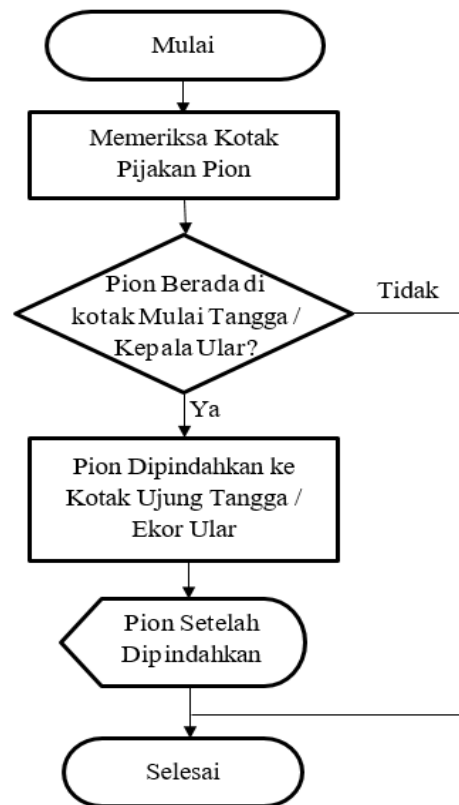
10. Perancangan Algoritma pion terkena ular atau tangga

Algoritma pion terkena ular atau tangga pada permainan ular tangga bergenre strategi dapat dilihat pada gambar 4.30. Pada gambar tersebut dijelaskan sistem akan memeriksa kotak yang dipijak pion. Ketika kotak yang dipijak pion ada kepala ular lawan atau kaki tangga sendiri, maka pion akan di pindahkan ke ekor ular atau ujung tangga, tetapi ketika kotak yang dipijak pion ada kepala ular sendiri atau kaki tangga lawan,

maka pion tidak akan dipindahkan. Dan ketika kotak yang dipijak pion tidak ada apa-apa, maka tidak akan terjadi apa-apa.



Gambar 4.29 Flowchart *Strategy phase*





Gambar 4.30 Flowchart pion terkena ular atau tangga

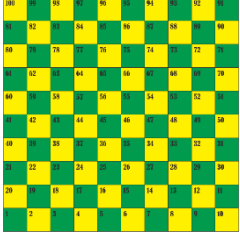



C. Perancangan Komponen Permainan

Perancangan komponen permainan meliputi deskripsi item yang akan muncul pada aplikasi permainan yang akan dibangun. Deskripsi item merupakan penjelasan dari setiap item yang muncul pada permainan. Berikut merupakan deskripsi item yang terdapat pada permainan ular tangga bergenre strategi seperti terlihat pada Tabel 4.3.

Tabel 4.3 Deskripsi Item

No.	Jenis Item	Deskripsi	Keterangan
1		Pion 1	Pion pemain 1
2		Pion 2	Pion pemain 2

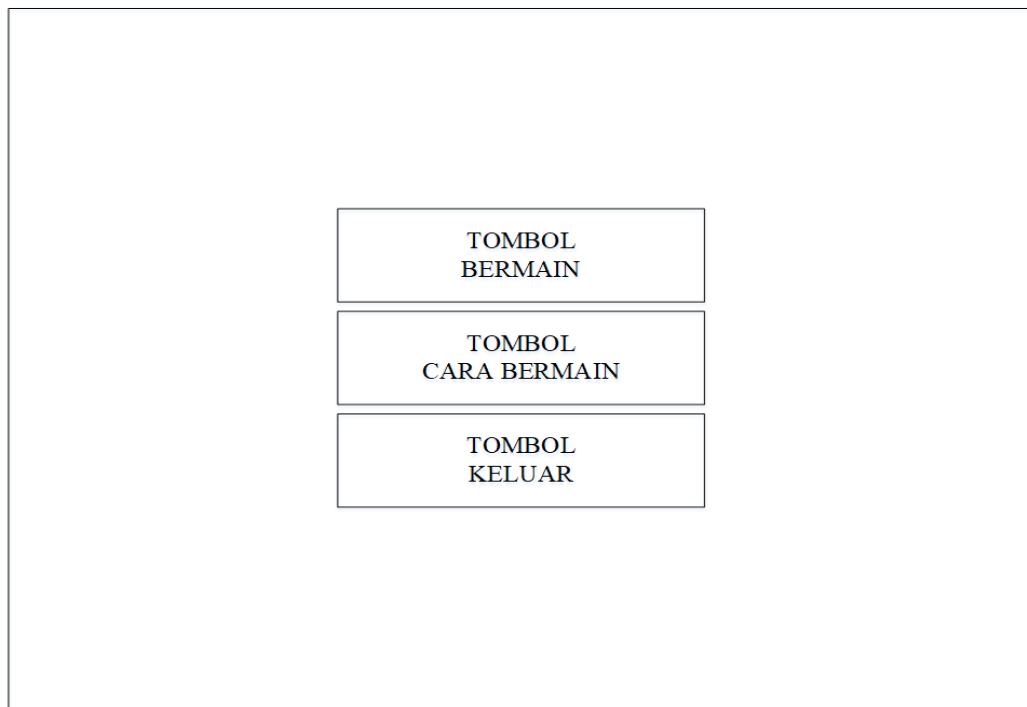
No.	Jenis Item	Deskripsi	Keterangan
3		Dadu	Untuk menentukan giliran pemaian dan untuk menentukan seberapa langkah tiap pion maju, dadu memiliki angka dari 1 samapai 6.
4		Kartu Level 1	Kartu level 1 berisi ular dan tangga, kartu bisa dibeli seharga 7 poin, dan dipasang bebas pada arena permainan, ada 12 jenis ular dan tangga yang berbeda-beda pada kartu level 1 ini. Kartu berjumlah 24.
5		Kartu Level 2	Kartu level 2 berisi ular dan tangga, kartu bisa dibeli seharga 10 poin, dan dipasang bebas pada arena permainan, ada 10 jenis ular dan tangga yang berbeda-beda pada kartu level 2 ini. Kartu berjumlah 16.
6		Kartu Level 3	Kartu level 3 berisi ular dan tangga, kartu bisa dibeli seharga 13 poin, dan dipasang bebas pada arena permainan, ada 18 jenis ular dan tangga yang berbeda-beda pada kartu level 3 ini. Kartu berjumlah 18.

No.	Jenis Item	Deskripsi	Keterangan
7		Board	Arena yang di gunakan untuk permainan
8		Ular	Ular adalah item untuk menurunkan pemain. Ular dipasang ketika mendapat kartu ular, ular warna merah berarti milik pemain 1, dan ular warna ungu berarti milik pemain 2. Setiap level memiliki panjang yang berbeda.
9		Tangga	Tangga adalah item untuk menaikkan pemaian. Tangga dipasang ketika mendapat kartu tangga, tangga warna merah berarti milik pemain 1, dan tangga warna ungu berarti milik pemain 2. Setiap level memiliki panjang yang berbeda.
10		Poin Bar	Poin bar adalah penanda poin yang di miliki setiap pemain, pemain 1 dan 2 memiliki poin bar masing-masing. Maksimal poin bar adalah 20.

d. Desain Antarmuka

Antarmuka adalah tampilan dari suatu perangkat lunak yang memiliki peran sebagai media komunikasi antara perangkat lunak dan pengguna. Perancangan ini merupakan sebuah penggambaran, perencanaan dan pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh. Perangkat lunak yang dibangun diharapkan menyediakan suatu tampilan antarmuka yang mudah dipahami dan digunakan oleh pengguna. Berikut ini adalah perancangan tampilan antarmuka yang digunakan pada permainan ular tangga bergenre strategi:

1. Perancangan Antarmuka Halaman Utama

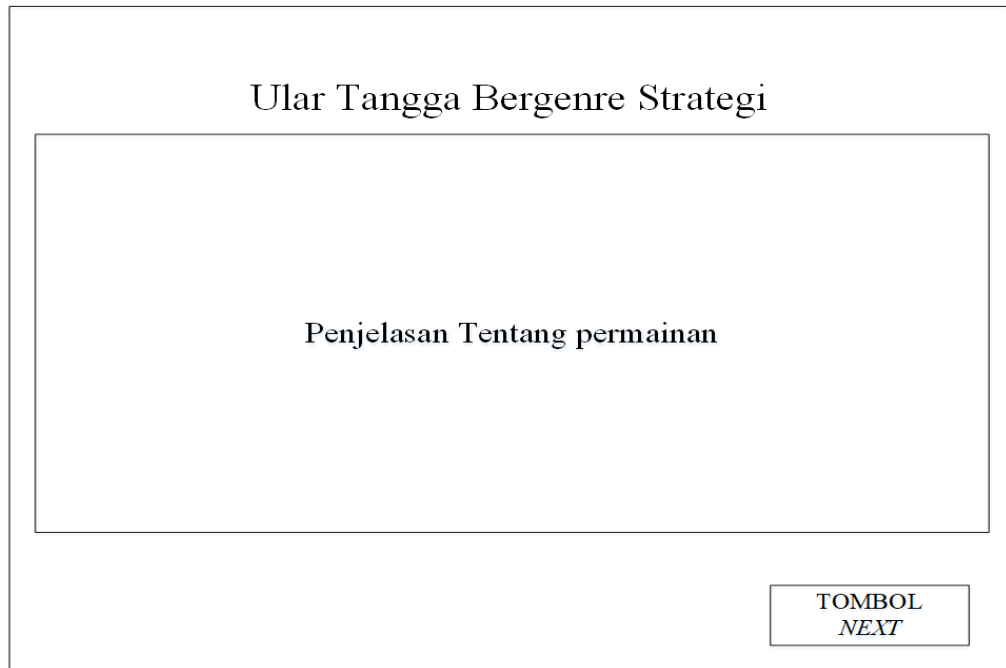


Gambar 4.31 Perancangan Antarmuka Halaman Utama

Gambar 4.31 menunjukkan perancangan antarmuka halaman utama. Halaman ini difungsikan sebagai halaman awal saat pertama kali aplikasi permainan ini dijalankan. Terdapat 3 tombol yaitu tombol dalam halaman menu utama yaitu tombol bermain yang berfungsi menuju halaman *preparation game*, tombol cara bermain yang berfungsi untuk menuju

halaman tentang permainan, dan tombol keluar yang berfungsi untuk keluar aplikasi.

2. Perancangan Antarmuka Halaman Penjelasan tentang Permainan

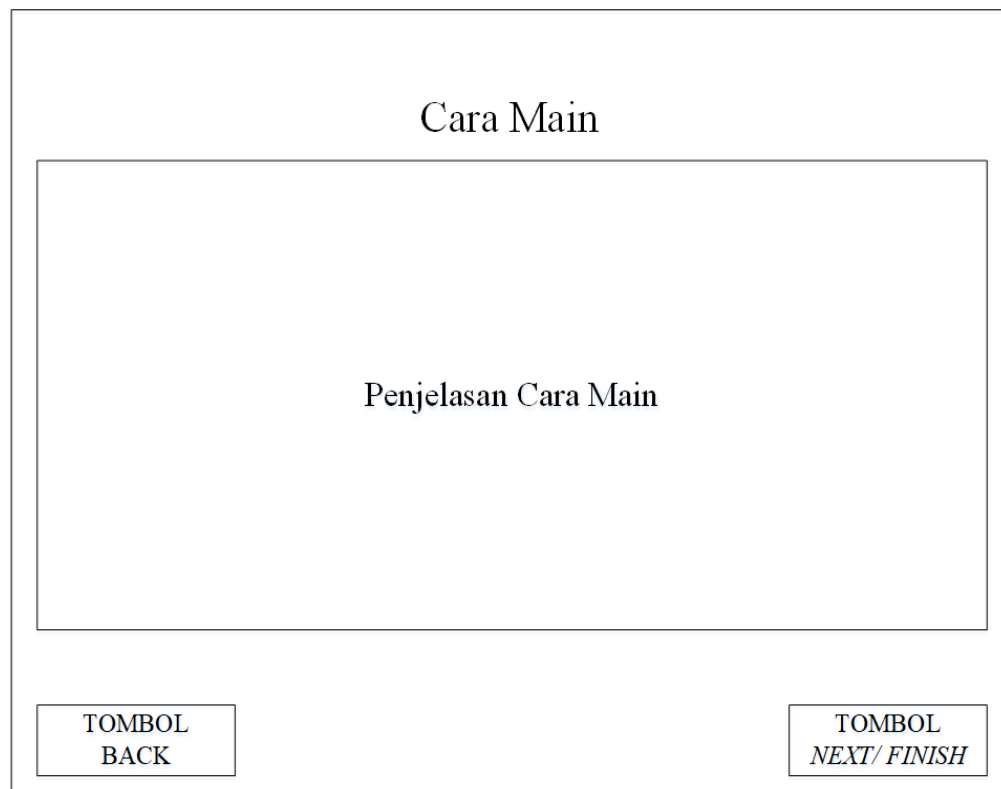


Gambar 4.32 Perancangan Antarmuka Halaman tentang Permainan

Gambar 4.32 menunjukkan perancangan antarmuka halaman tentang Permainan. Halaman ini difungsikan sebagai halaman menjelaskan tentang permainan ular tangga berggenre strategi. Terdapat 1 tempat *text* yang berisi tentang penjelasan permainan dan 1 tombol *next* yang berfungsi menuju halaman Cara bermain.

3. Perancangan Antarmuka Halaman Cara Bermain

perancangan antarmuka halaman cara bermain dapat dilihat pada gambar 4.33. Halaman ini difungsikan sebagai halaman untuk menjelaskan cara bermain permainan ular tangga berggenre strategi ini. Terdapat 1 tempat untuk memutar video cara bermain dan ada 2 tombol back dan tombol finish. Tombol back berfungsi menuju halaman Tentang permainan sedangkan untuk tombol finish berfungsi menuju halaman menu utama.



Gambar 4.33 Perancangan Antarmuka Halaman Cara Bermain

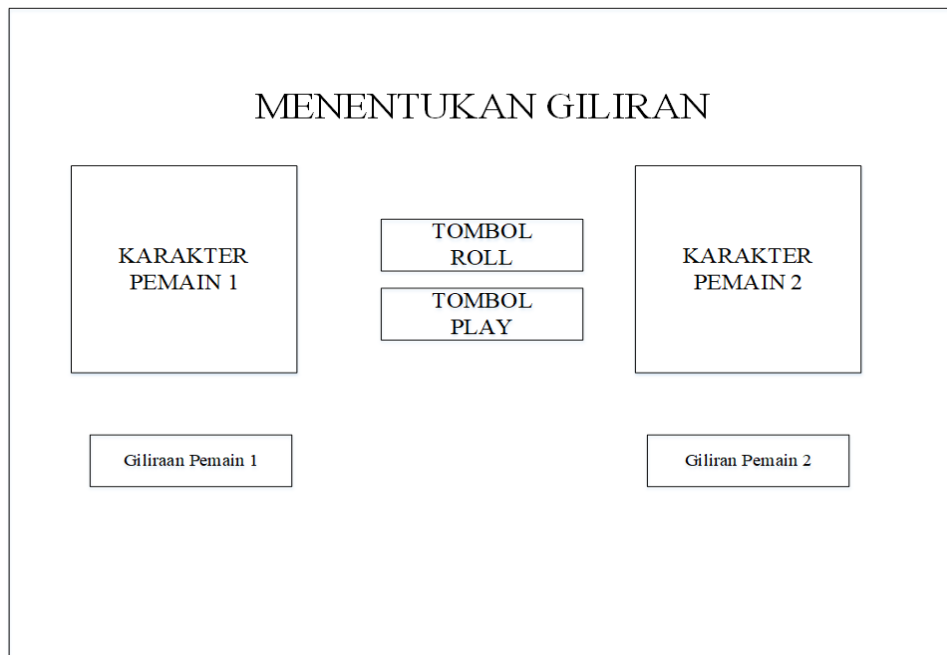
4. Perancangan Antarmuka Halaman *Preparation Game*

Pada gambar 4.34 menunjukkan perancangan antarmuka halaman *preparation game*. Halaman ini difungsikan sebagai untuk menentukan giliran pemain. Terdapat 2 tempat text untuk menampilkan giliran yang didapat pemain, 2 gambar untuk karakter pemain, dan 2 tombol. 1 tombol roll yang berfungsi untuk menentukan giliran pemain dan tombol *play* yang berfungsi untuk menuju halaman permainan.

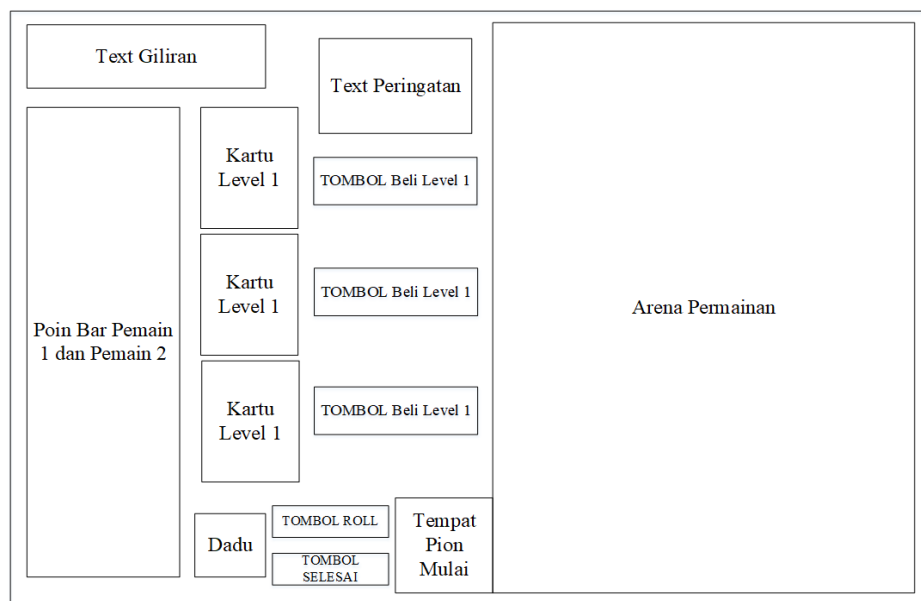
5. Perancangan Antarmuka Halaman Permainan

Perancangan antarmuka halaman permainan dapat dilihat pada gambar 4.35. Halaman ini difungsikan sebagai tempat permainan dilakukan. Terdapat 1 tempat *text* yang berisi giliran pemain, 1 tempat *text* yang berisi peringatan ketika item ada masalah, 1 tempat untuk poin bar pemain 1 dan pemain 2, 1 tempat untuk papan permainan(board), 3 tempat untuk tempat kartu level 1, level 2, level 3, 1 tempat untuk dadu, 1 tempat untuk kotak start, dan 5 tombol, tombol *roll* untuk untuk mengacak dadu, tombol

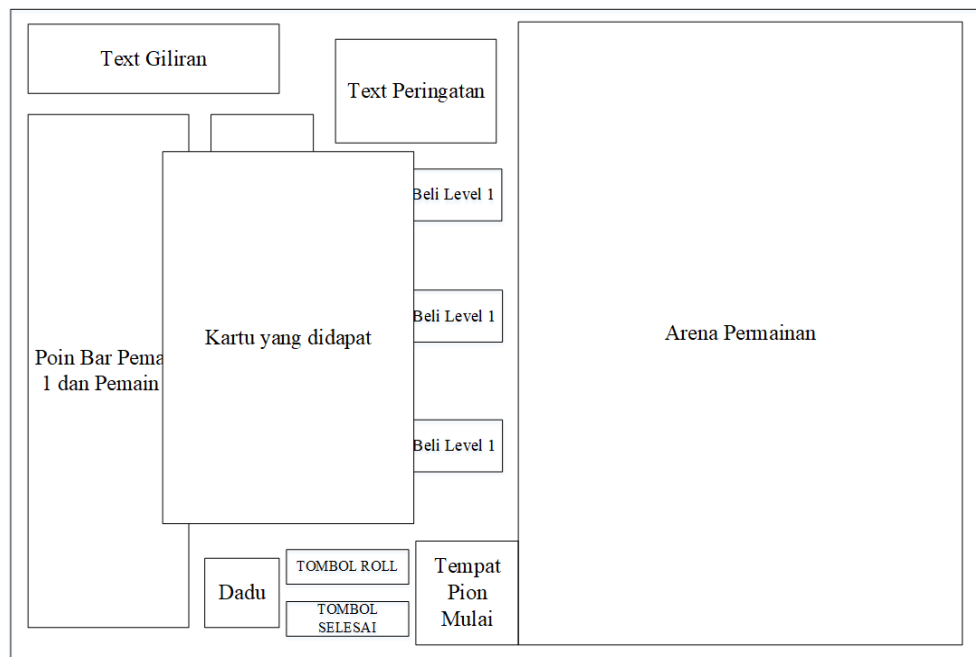
selesai untuk mengakhiri giliran, tombol beli level 1 untuk membeli kartu level 1, tombol beli level 2 untuk membeli kartu level 2, dan tombol beli level 3 untuk membeli kartu level 3. Ketika pemain membeli kartu, maka ada 1 tempat lagi untuk kartu yang didapat pemain (*card on hand*). Seperti yang terlihat pada gambar 4.36.



Gambar 4.34 Perancangan Antarmuka Halaman *Preparation Game*

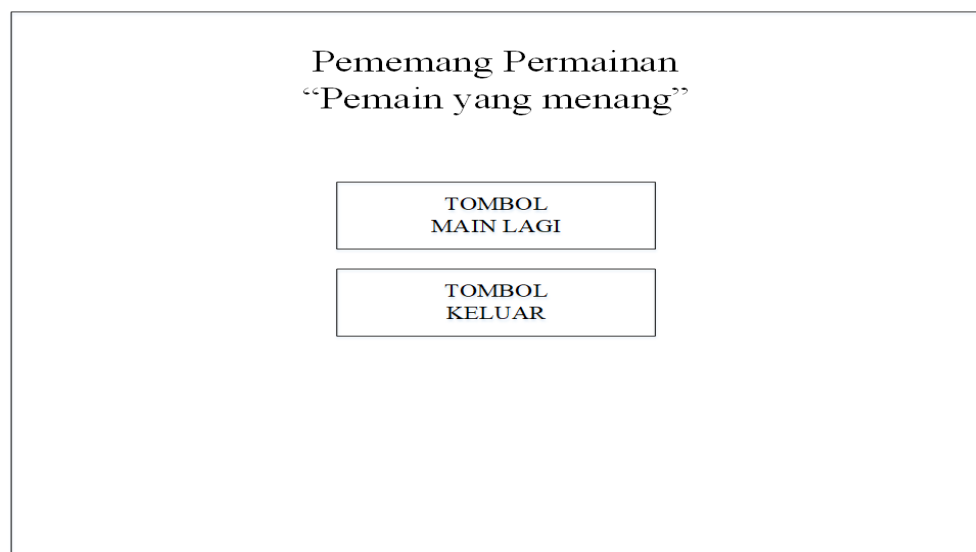


Gambar 4.35 Perancangan Antarmuka Halaman Permainan



Gambar 4.36 Perancangan Antarmuka Halaman Permainan saat membeli kartu

6. Perancangan Antarmuka Halaman Pemenang



Gambar 4.37 Perancangan Antarmuka Halaman Pemenang

Pada gambar 4.37 menunjukkan perancangan antarmuka halaman pemenang. Halaman ini difungsikan sebagai halaman untuk menunjukan pemenang dari permainan. Terdapat 1 *text* untuk menampilkan pemenang dan terdapat 3 tombol yaitu Tombol Main Lagi yang berfungsi menuju halaman *starting player* dan Tombol Keluar yang berfungsi untuk keluar aplikasi.

4.2.3 Hardware dan Software Pengguna

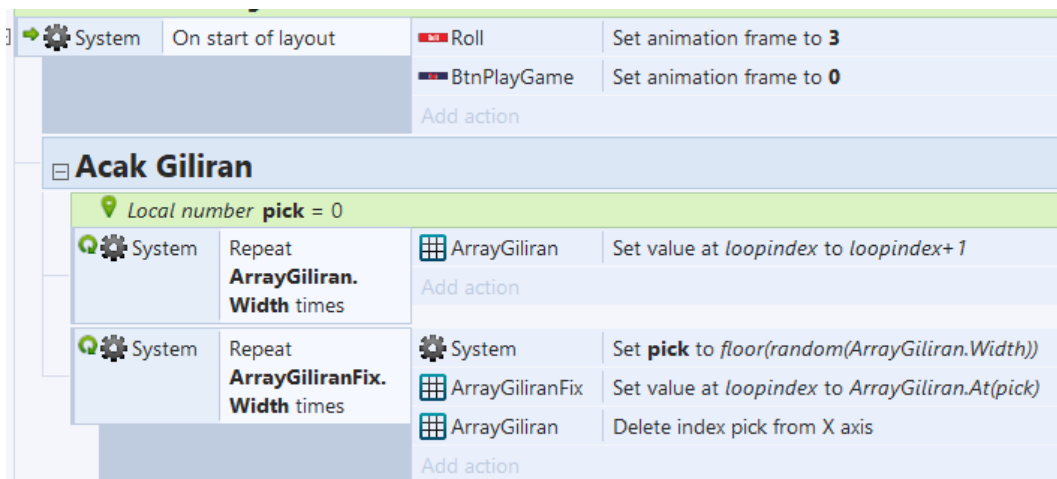
Karena kurangnya literatur atau informasi mengenai spesifikasi *hardware* dan *software* pengguna, maka hal ini akan menjadi salah satu bagian yang diujikan pada penelitian ini.

4.2.4 Implementasi fungsi

Perancangan fungsi dilakukan berdasarkan hasil dari tahap analisa. Fungsi yang diperlukan untuk menjalankan aplikasi permainan ular tangga bergenre strategi adalah sebagai berikut:

a. Event *Preparation Game*

Event *Preparation Game* dibuat berdasarkan pada sub-bab 4.2.2 poin b nomor 5. Event ini digunakan pada saat awal ketika pemain memilih menu main, event ini berfungsi untuk menentukan giliran pemain. *Condition Commands* yang digunakan pada event ini adalah *On start of layout*, *Compare two values*, *clicked*, dan *else*. *Action commands* pada event ini adalah *set animation frame*, *play*, *set value*, *set text*, dan *go to layout*. Dan ada 2 variabel digunakan yaitu *AngkaGiliranP1* dan *AngkaGiliranP2*. Dan ada 2 array yang berisikan angka giliran yang belum diacak dan angka giliran yang sudah diacak.



Gambar 4.38 Source code On start of layout pada event preparation game

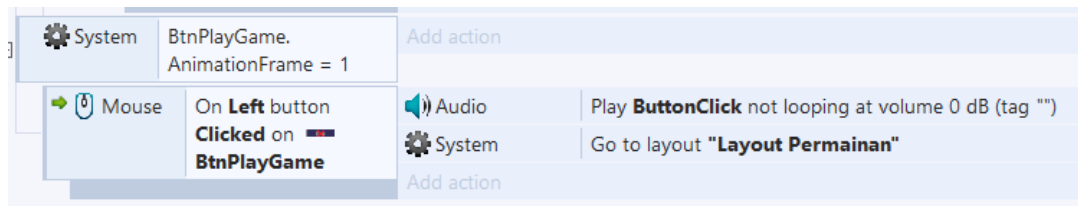
Gambar 4.38 adalah gambar *source code* yang dijalankan pertama setelah *layout starting player* mulai ditampilkan, *Condition Commands* yang dijalankan

setelah *layout starting player* ditampilkan adalah *On start of layout*, *On start of layout* akan menjalankan action *set animation frame* milik Tombol Roll menjadi *frame* yang ke-3 agar bisa tekan dan Tombol BtnPlayGame menjadi *frame* yang ke-0 agar tombol BtnPlayGame tidak bisa ditekan. Serta mengacak array yang sudah disiapkan untuk menentukan giliran.



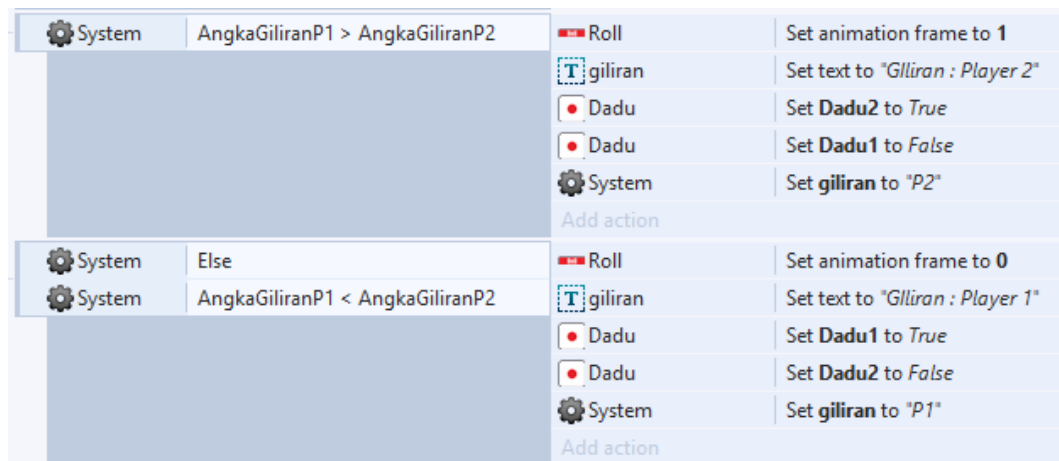
Gambar 4.39 Source code ketika tombol roll ditekan pada *preparation game*

Gambar 4.39 adalah *source code* setelah *On start of layout* dijalankan, ketika Tombol Roll ditekan pada saat *frame* sama dengan 3, maka action yang dijalankan adalah *set variable* AngkaGiliranP1 berdasarkan nilai array yang diacak pada *index* 0, *set variable* AngkaGiliranP2 berdasarkan nilai array yang diacak pada *index* 1, *play* efek suara tombol, *set animation frame* Tombol Roll menjadi *frame* yang ke-2 agar tidak bisa tekan dan Tombol BtnPlayGame menjadi animasi yang ke-1 agar tombol ini bisa ditekan, dan sistem akan menampilkan hasil giliran dan menampilkan yang mendapat giliran pertama.



Gambar 4.40 Source code ketika tombol BtnPlayGame ditekan pada *preparation game*

Gambar 4.40 adalah *source code* setelah Hasil giliran pemain sudah muncul dan tombol BtnPlayGame sudah diaktifkan. Setelah *frame* pada BtnPlayGame menjadi 1/aktif, maka pemain harus menekan tombol BtnPlayGame untuk bermain permainannya, ketika pemain menekan tombol BtnPlayGame, sistem akan menjalankan efek suara tombol dan sistem memindahkan ke *frame* permainan dengan perintah *go to layout*.

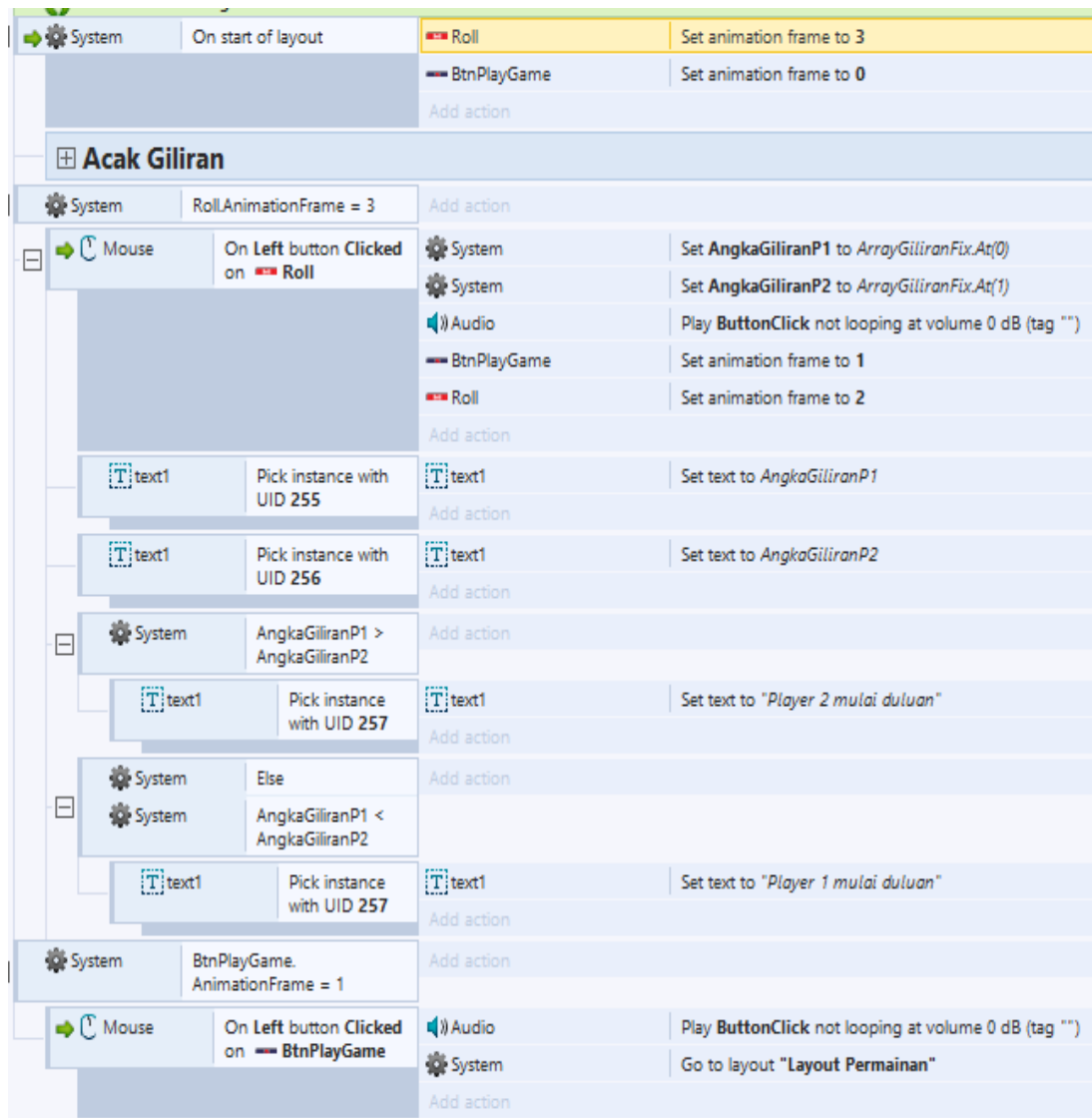


Gambar 4.41 Source code menjalankan hasil dari *preparation game* pada *layout* permainan

Supaya *layout* permainan menjalankan sesuai dengan hasil pada *starting player* setelah pindah *layout* ke *layout* permainan. Maka dibuatlah *source code* seperti pada gambar 4.41. Pada gambar 4.41, *Variable* AngkaGiliranP1 dengan *variable* AngkaGiliranP2 dibandingkan. *Action* yang dijalankan ketika AngkaGiliranP1 lebih besar dari AngkaGiliranP2 adalah BtnRoll *set animation frame* menjadi 0, *set text* menjadi giliran player 1, *set boolean* Dadu1 pada objek Dadu menjadi true, *set boolean* Dadu2 pada objek Dadu menjadi false, dan *set variable* giliran menjadi "P1" agar menandakan bahawa itu giliran Player 1. Dan ketika AngkaGiliranP1 lebih kecil dari AngkaGiliranP2 *action* yang dijalankan adalah BtnRoll *set animation frame*

menjadi 1, *set text* menjadi giliran player 2, *set boolean* Dadu1 pada objek Dadu menjadi *false*, *set boolean* Dadu2 pada objek Dadu menjadi *true*, dan *set variable* giliran menjadi “P2” agar menandakan bahwa itu giliran Player 2.

Source code keseluruhan event *preparation game* dapat dilihat pada gambar 4.42 dan 4.43. dan tampilan event *preparation game* dapat dilihat pada gambar 4.44 dan 4.45.



Gambar 4.42 *Source code* Event *preparation game* pada *layout preparation game*

System	AngkaGiliranP1 > AngkaGiliranP2	Roll	Set animation frame to 1
		giliran	Set text to "Giliran : Player 2"
		Dadu	Set Dadu2 to True
		Dadu	Set Dadu1 to False
		System	Set giliran to "P2"
		Add action	
System	Else	Roll	Set animation frame to 0
System	AngkaGiliranP1 < AngkaGiliranP2	giliran	Set text to "Giliran : Player 1"
		Dadu	Set Dadu1 to True
		Dadu	Set Dadu2 to False
		System	Set giliran to "P1"
		Add action	

Gambar 4.43 Source code Event preparation game Pada layout permainan



Gambar 4.44 Tampilan sebelum melakukan preparation game

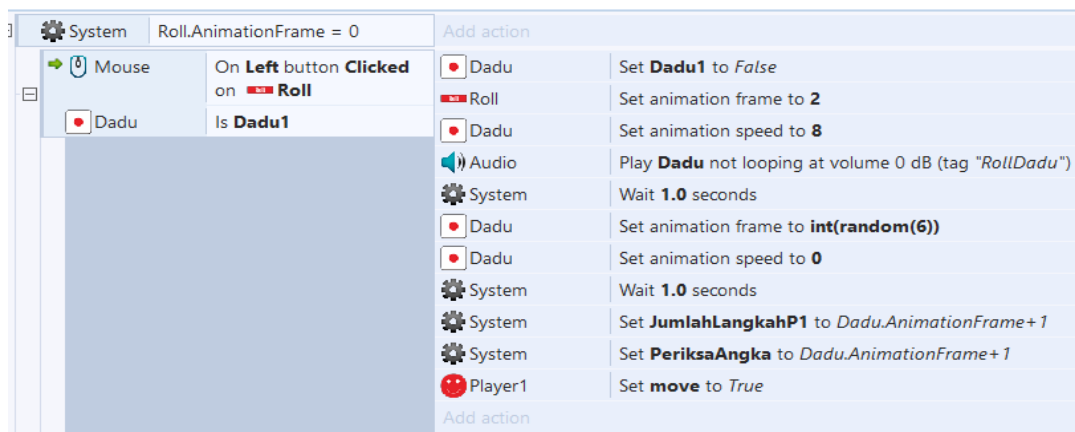


Gambar 4.45 Tampilan setelah melakukan preparation game

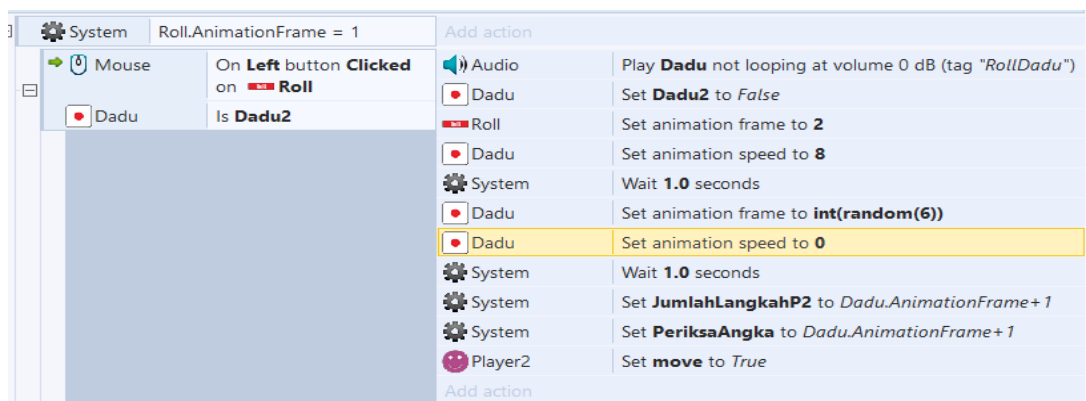
b. Event Roll Dadu Pada Saat Permainan

Event Roll dadu dibuat berdasarkan pada sub bab 4.2.2 bagian b point 7, event ini berada pada *turn phase*, event ini berfungsi untuk menentukan angka dadu yang didapat pemain untuk menjalankan pion dan berapa jumlah poin yang ditambahkan pada poin bar pemain yang melakukan roll.

Condition Commands yang digunakan pada event ini adalah *Compare two values*, *compare variable*, *clicked*, *is boolean*, dan *else*. *Action commands* pada event ini adalah *set boolean*, *set animation frame*, *set value*, dan *wait*. Dan ada 3 *variable* yang digunakan pada event ini yaitu Jumlah Langkah Player, Periksa angka, dan periksa angka 6 yang ke berapa. Pada saat *turn phase* event ini akan difungsikan pertama kali.



Gambar 4.46 Source code ketika tombol roll ditekan pada *turn phase* (Pemain 1)

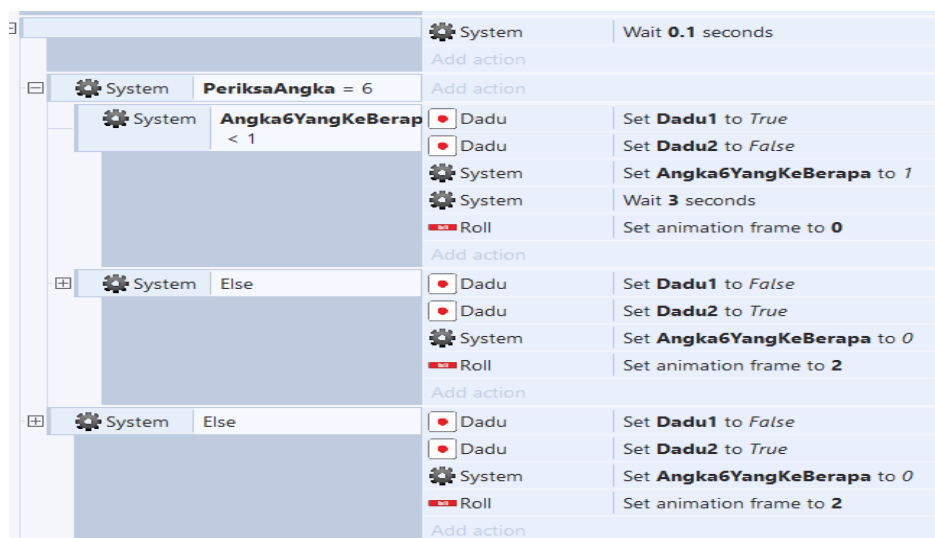


Gambar 4.47 Source code ketika tombol roll ditekan pada *turn phase* (Pemain 2)

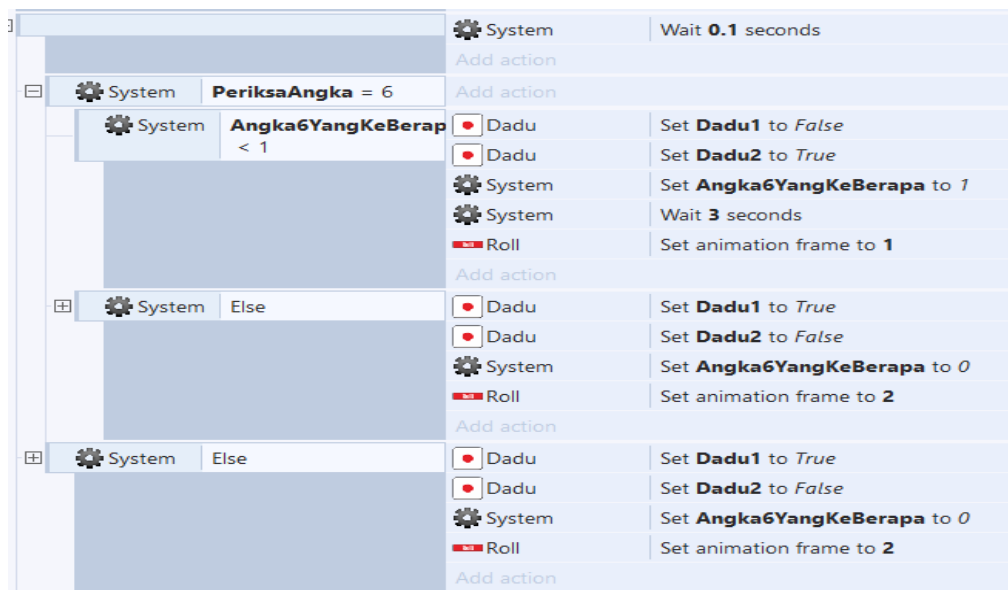
Pada gambar 4.46 dan 4.47 adalah gambar *source code* tombol roll ketika ditekan pada saat *turn phase*, *source code* ini yang pertama kali

dijalankan pada event ini. Pada gambar dijelaskan, hal yang pertama adalah sistem akan memeriksa *frame* tombol roll yang aktif dengan *compare two values*. Jika *frame* pada tombol roll sama dengan 0 berarti giliran pemain 1, dan proses selanjutnya akan menjalankan bagian-bagian untuk pemain 1, begitu juga untuk player 2, jika *frame* pada tombol roll sama dengan 1 berarti itu giliran pemain 2, dan proses selanjutnya akan menjalankan bagian-bagian untuk pemain 2.

Ketika *frame* sudah diperiksa oleh sistem, selanjutnya pemain dipersilahkan untuk menekan tombol roll. Ketika pemain menekan tombol roll, dan *boolean* dadu bernilai true baik dadu1 untuk pemain 1 dan dadu2 untuk pemain 2, sistem akan menjalankan *action set boolean* dadu1 atau dadu2 dibuat *false* agar action ini tidak berjalan 2 kali. *set animation frame* tombol roll ke animasi ke-3, agar tombol roll tidak bisa ditekan pada proses ini berjalan, *set animation speed* dibuat 8 untuk animasi acak dadu. Setelah satu detik, *set animation speed* dibuat 0 kembali agar animasi tidak mengacak lagi. *set animation frame* pada dadu dibuat “*int(random(6))*” agar angka dadu yang dimunculkan adalah acak. *Set value* pada *variable* jumlah langkah dan periksa angka menjadi animasi yang ditampilkan oleh dadu. Serta *set boolean* move pada pion dibuat true, agar pion berjalan/ bergerak.



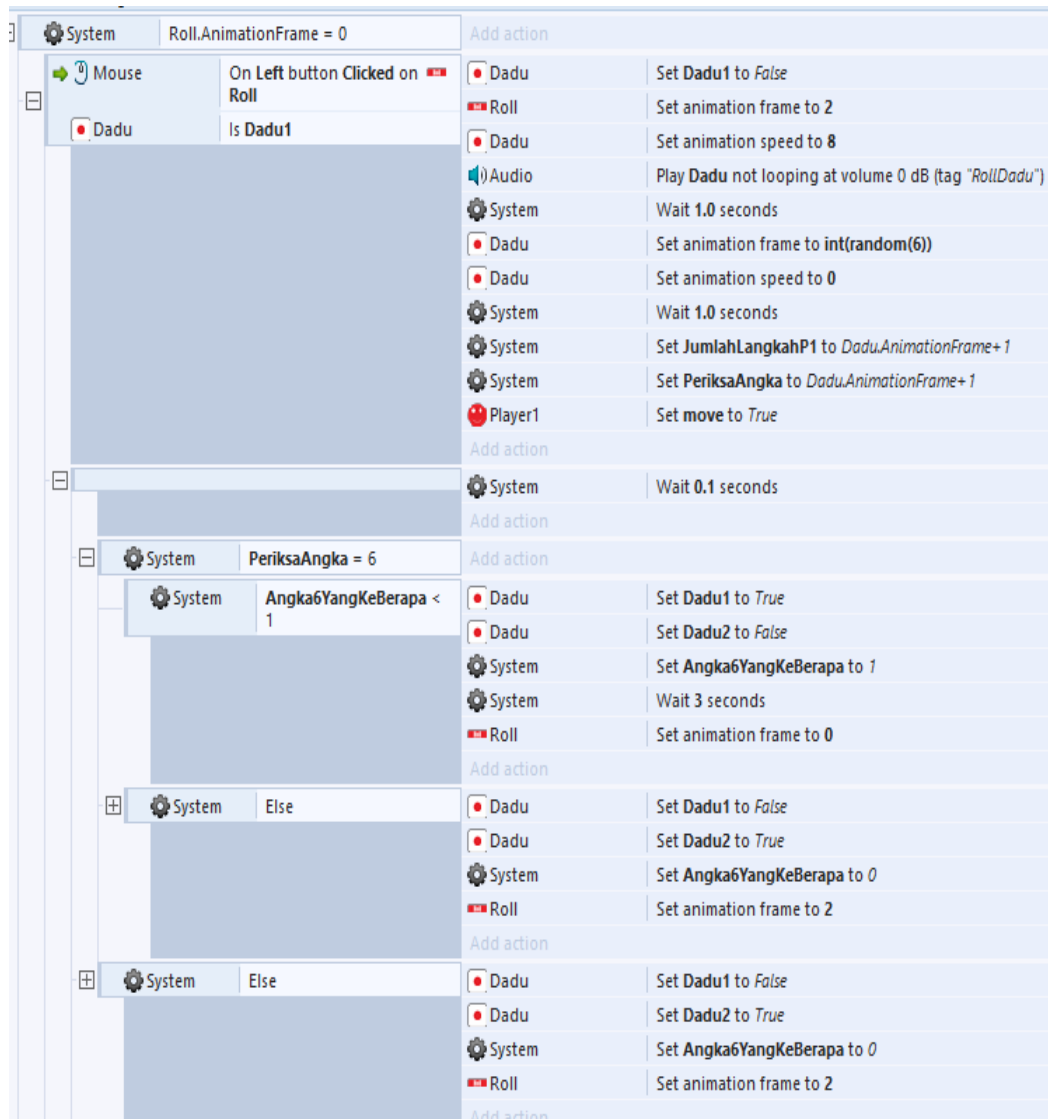
Gambar 4.48 Source code menentukan bonus *turn* pada *turn phase* (Pemain 1)



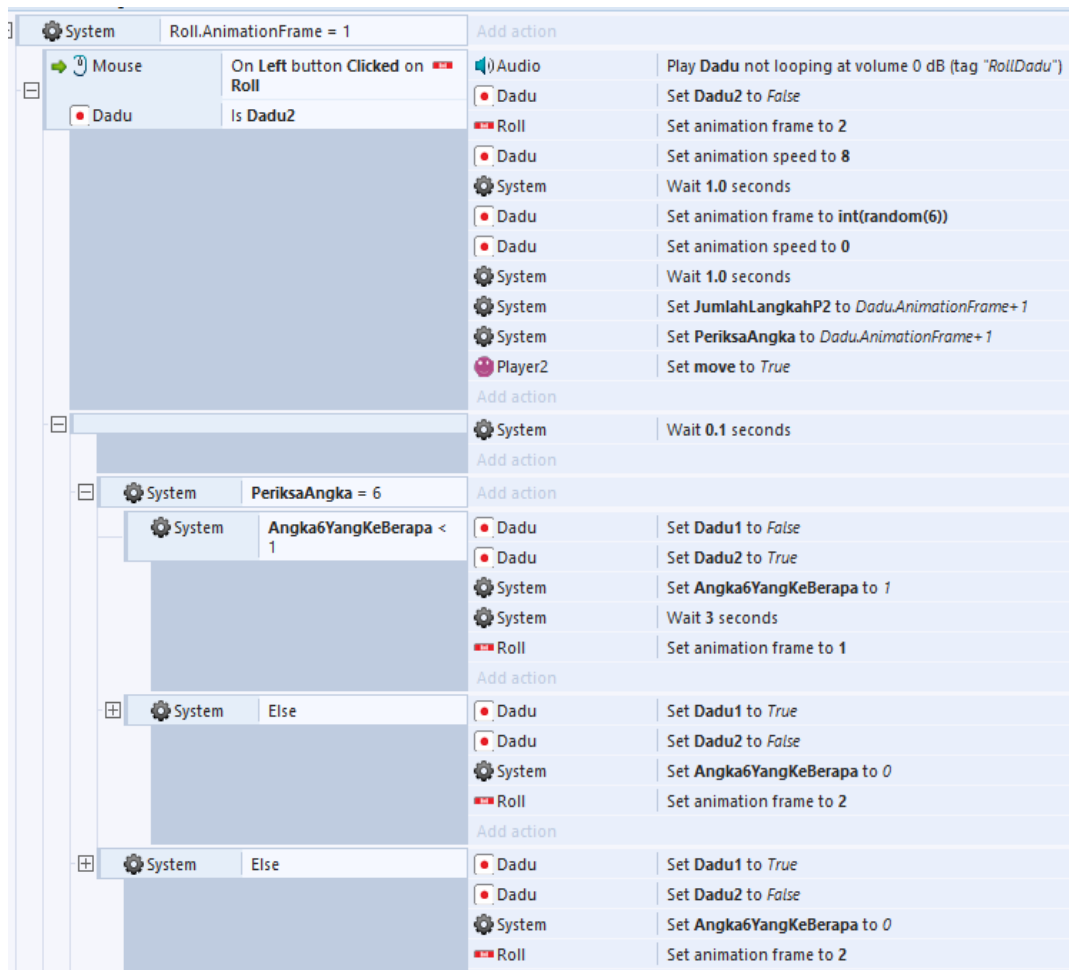
Gambar 4.49 Source code menentukan bonus *turn* pada *turn phase* (Pemain 2)

Pada gambar 4.48 dan 4.49 menjelaskan *source code* proses pemeriksaan apakah pemain 1 atau pemain 2 mendapat bonus *turn* atau tidak. Setelah proses pengacakan dan angka dadu sudah muncul, sistem dibuat menunggu 0.1 detik agar bisa menjalankan proses selanjutnya. Setelah itu sistem akan memeriksa angka dadu yang didapat dengan cara *compare variable* periksa angka. Jika angka *variable* yang di *compare* tidak sama dengan 6, maka sistem akan menjalankan action *set boolean* Dadu menjadi giliran selanjutnya. Dan *set animation frame* pada tombol roll menjadi 2 agar tidak bisa di tekan dan setelah itu menuju event menggerakkan pion. Dan jika angka *variable* yang di *compare* sama dengan 6, maka sistem akan memeriksa *variable* angka 6 yang ke berapa dengan cara *compare variable*. Jika *variable* angka 6 ke berapa kurang dari 1, maka sistem akan mengulangi proses event ini dari awal dengan menambahkan angka 1 pada *variable* angka 6 ke berapa. Dan Jika *variable* angka 6 ke berapa lebih dari 0 maka sistem akan menjalankan action *set boolean* Dadu menjadi giliran selanjutnya. *Set variable* angka 6 yang ke berapa menjadi 0, Dan *set animation frame* pada tombol roll menjadi 2 agar tidak bisa di tekan dan setelah itu menuju event menggerakkan pion.

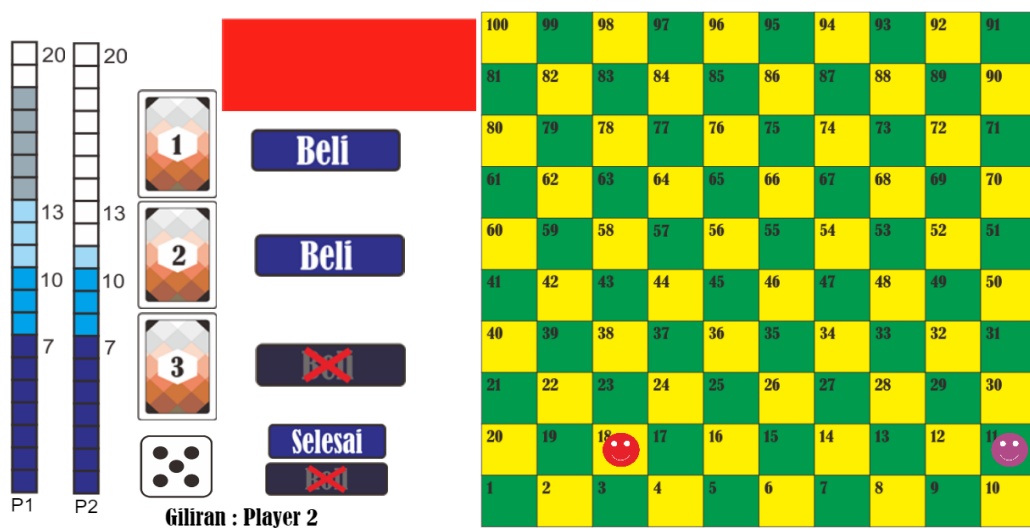
Source code keseluruhan event roll dadu saat permainan dapat dilihat pada gambar 4.50 dan 4.51. Dan tampilan event roll dadu saat permainan dapat dilihat pada gambar 4.52 dan 4.53.



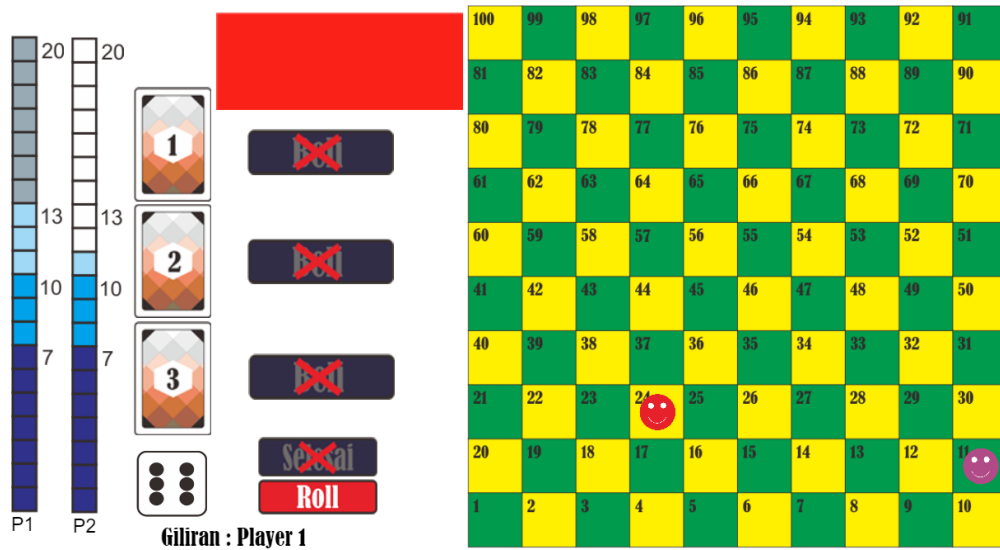
Gambar 4.50 *Source code* event roll dadu pada saat permainan (pemain 1).



Gambar 4.51 Source code event roll dadu pada saat permainan (pemain 2).



Gambar 4.52 Tampilan sebelum roll dadu



Gambar 4.53 Tampilan setelah roll dadu

c. Event Mengacak Kartu

Event mengacak kartu adalah event untuk mengacak kartu di setiap level agar urutan kartu di setiap permainan berbeda-beda, event ini dijalankan setelah *layout* permainan ditampilkan. Event ini bagian dari *starting game*.

Condition Commands yang digunakan pada event ini adalah *Repeat*. *Action commands* pada event ini adalah *set value* dan *delete*, 1 *variable local* yaitu *pick*, dan 2 array untuk menyimpan urutan angka keberadaan kartu sebelum diacak dan urutan angka kartu setelah kartu diacak.

Gambar 4.54 menjelaskan *Source code* ketika *layout* permainan ditampilkan sistem akan menjalankan *Condition Commands Repeat* sejumlah panjang array untuk mengisi array urutan angka keberadaan kartu sebelum diacak dengan angka yangurut dari 1 sampai sejumlah kartu. Kemudian sistem akan menjalankan *Condition Commands Repeat* yang kedua untuk mengisi urutan angka kartu setelah kartu diacak dengan cara setiap perulangan menjalankan *Action commands set value* pada *variable pick* dengan *index* yang diacak. Setelah *variable pick* terisi dengan *index* yang diacak, sistem akan mengisi array yang kedua dengan isi yang ada pada array pertama pada *index* yang bernilai *pick*. Setelah array kedua sudah menyimpannya, *index* yang bernilai *pick* pada array yang pertama dihapus, agar nilai array pertama yang

sudah tersimpan pada array kedua tidak keluar lagi pada perulangan selanjutnya.

The image shows a source code editor with three sections for shuffling cards at different levels:

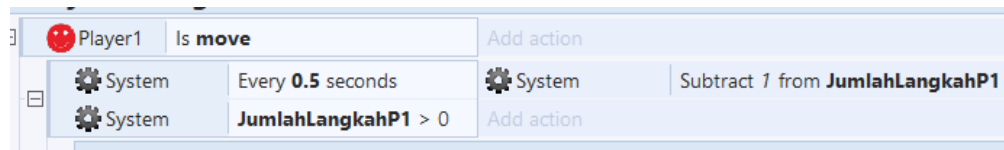
- Acak Kartu lvl1:**
 - Local number **pick = 0**
 - Repeat **KartuLvl1.Width** times:
 - KartuLvl1: Set value at *loopindex* to *loopindex+ 1*
 - System: Set **pick** to $\text{floor}(\text{random}(\text{KartuLvl1.Width}))$
 - KartuLvl1fix: Set value at *loopindex* to $\text{KartuLvl1.At}(\text{pick})$
 - KartuLvl1: Delete index pick from X axis
- Acak Kartu lvl2:**
 - Local number **pick = 0**
 - Repeat **KartuLvl2.Width** times:
 - KartuLvl2: Set value at *loopindex* to *loopindex+ 1*
 - System: Set **pick** to $\text{floor}(\text{random}(\text{KartuLvl2.Width}))$
 - KartuLvl2fix: Set value at *loopindex* to $\text{KartuLvl2.At}(\text{pick})$
 - KartuLvl2: Delete index pick from X axis
- Acak Kartu lvl3:**
 - Local number **pick = 0**
 - Repeat **KartuLvl3.Width** times:
 - KartuLvl3: Set value at *loopindex* to *loopindex+ 1*
 - System: Set **pick** to $\text{floor}(\text{random}(\text{KartuLvl3.Width}))$
 - KartuLvl3fix: Set value at *loopindex* to $\text{KartuLvl3.At}(\text{pick})$
 - KartuLvl3: Delete index pick from X axis

Gambar 4.54 Source code mengacak kartu level 1, level 2, dan level 3

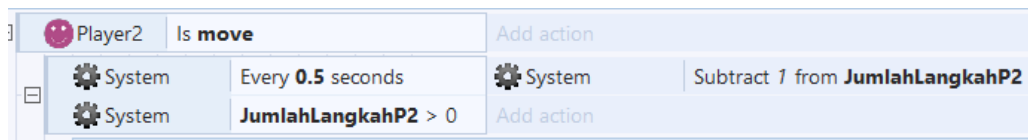
d. Event Menggerakan Pion

Event menggerakan pion ini berada pada *turn phase*, event ini dijalankan setelah event roll dadu. Event ini dibuat berdasarkan pada sub bab 4.2.2 bagian b point 7. Event ini juga berfungsi untuk menggerakan animasi pion berjalan pada kotak yang dituju sesuai angka dadu yang didapat.

Condition Commands yang digunakan pada event ini adalah *Every X second*, *Compare variable*, *is boolean*, dan *else*. *Action commands* pada event ini adalah *subtract from*, *set boolean*, *set value*, *add value*, dan *GirdMove*. Dan ada 4 *variable* yang digunakan pada event ini yaitu Jumlah Langkah Player, Direction, PosisiX, dan PosisiY.



Gambar 4.55 Source code mengurangi langkah pemain 1



Gambar 4.56 Source code mengurangi langkah pemain 2

Setelah event roll dadu pada saat permainan selesai. Sistem akan menjalankan *source code* pada gambar 4.55 dan 4.56. Gambar 4.55 dan 4.56 menjelaskan sistem akan mengecek *boolean* move bernilai true atau false, ketika *boolean* move bernilai true, setiap 0,5 detik ketika *variable* jumlah langkah > 0, sistem akan mengurangi *variable* jumlah langkah senilai 1.



Gambar 4.57 Source code menentukan variable direction untuk pemain 1



Gambar 4.58 *Source code* menentukan *variable direction* untuk pemain 2

Gambar 4.57 dan 4.58 adalah *source code* setelah *variable* langkah pemain sudah dikurangi. Setelah dikurangi sistem akan memeriksa pion apakah sudah pernah di kotak 100, ketika pion sudah pernah di kotak 100 maka sistem akan langsung *set variable* *direction* menjadi kanan. Namun jika pion belum menyentuh kotak 100 maka sistem akan menentukan *variable* *direction* berdasarkan nilai *variable* posisiY dari pion. Ketika *variable* posisiY pion bernilai genap maka *variable* *direction*-nya di set menjadi kiri dan ketika posisiY pion bernilai Ganjil maka *variable* *direction*-nya di set menjadi kanan.

System	DirectionP1 = "kiri"	Add action
System	PosisiP1x = 1	Add -1 to PosisiP1y
System	Else	Add -1 to PosisiP1x
System	DirectionP1 = "kanan"	Add action
System	PosisiP1x = 10	Add -1 to PosisiP1y
System	Else	Add 1 to PosisiP1x
Player1	GridMove move to [PosisiP1x, PosisiP1y]	
Audio	Play jump not looping at volume 0 dB (tag "jalan")	

Gambar 4.59 *Source code* setelah *variable direction* untuk pemain 1 sudah ditentukan

System	DirectionP2 = "kiri"	Add action
System	PosisiP2x = 1	Add -1 to PosisiP2y
System	Else	Add -1 to PosisiP2x
System	DirectionP2 = "kanan"	Add action
System	PosisiP2x = 10	Add -1 to PosisiP2y
System	Else	Add 1 to PosisiP2x
Player2	GridMove move to [PosisiP2x, PosisiP2y]	
Audio	Play jump not looping at volume 0 dB (tag "jalan")	

Gambar 4.60 *Source code* setelah *variable direction* untuk pemain 2 sudah ditentukan

Pada gambar 4.59 dan 4.60 menjelaskan setelah *variable direction* sudah terisi/ditentukan, selanjutnya sistem akan memeriksa *variable* yang didapatkan yaitu kanan atau kiri, ketika *variable direction* sama dengan kanan, maka sistem selanjutnya memeriksa *variable* posisiX dari pion. ketika *variable* posisiX pion sama dengan 10 maka sistem akan mengurangi *variable* posisiY senilai 1, namun ketika *variable* posisiX pion tidak sama dengan 10 maka sistem akan menambahkan *variable* posisiX senilai 1. Dan ketika *variable direction* sama dengan kiri, maka sistem selanjutnya memeriksa *variable* posisiX dari pion. Ketika *variable* posisiX pion sama dengan 1 maka sistem akan mengurangi *variable* posisiY senilai 1. dan ketika *variable* posisiX pion tidak sama dengan 1 maka sistem akan mengurangi *variable* posisiX senilai 1. setelah itu sistem memindahkan pion ke posisi *variable* posisiX dan posisiY yang baru.

System	JumlahLangkahP1 = 0	Player1	Set move to <i>False</i>
		Player1	Set Sudah100 to <i>False</i>
		Add action	
System	JumlahLangkahP2 = 0	Player2	Set move to <i>False</i>
		Player2	Set Sudah100 to <i>False</i>
		Add action	

Gambar 4.61 *Source code* Ketika jumlah langkah pemain = 0

Proses ini akan terus berjalan sampai *variable* jumlah langkah player menjadi 0, ketika *variable* jumlah langkah player bernilai 0 maka sistem akan *set boolean* move menjadi *false* dan *boolean* sudah100 menjadi *false* seperti yang dijelaskan *source code* pada gambar 4.61.

Source code keseluruhan event menggerakkan pion dapat dilihat pada gambar 4.62 dan 4.63. Dan tampilan event Menggerakkan pion dapat dilihat pada gambar 4.64 dan 4.65.

Player 1 Bergerak

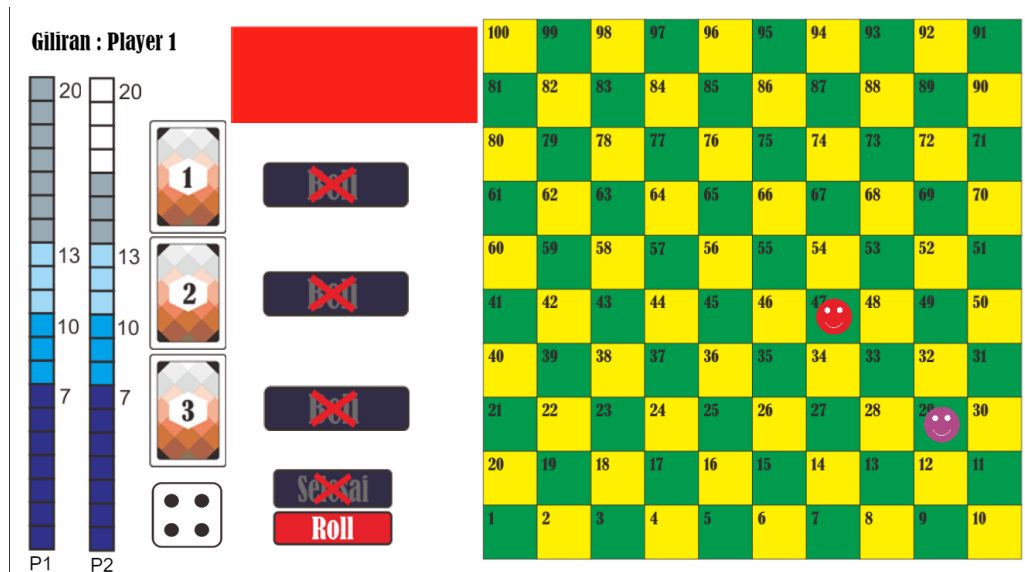
Player1	Is move	Add action
System	Every 0.5 seconds	System Subtract 1 from JmlLangkahP1
System	JmlLangkahP1 > 0	Add action
Menambahkan poin P1		
System	PosisiP1x = 1	Player1 Set Sudah100 to True
System	PosisiP1y = 0	Add action
Player1	Is Sudah100	System Set DirectionP1 to "kanan"
		Add action
System	Else	System Set NilaiY to PosisiP1y%2
		Add action
Syst	NilaiY = 0	System Set DirectionP1 to "kiri"
		Add action
Syst	Else	System Set DirectionP1 to "kanan"
		Add action
System	DirectionP1 = "kiri"	Add action
Syst	PosisiP1x = 1	System Add -1 to PosisiP1y
		Add action
Syst	Else	System Add -1 to PosisiP1x
		Add action
System	DirectionP1 = "kanan"	Add action
Syst	PosisiP1x = 10	System Add -1 to PosisiP1y
		Add action
Syst	Else	System Add 1 to PosisiP1x
		Add action
Player1	GridMove move to [PosisiP1x, PosisiP1y]	Add action
Audio	Play jump not looping at volume 0 dB (tag "jalan")	Add action
System	JmlLangkahP1 = 0	System Set faseGiliran to giliran
Dadu	Is Dadu1	Add action

Gambar 4.62 Source code event menggerakkan pion(pemain 1).

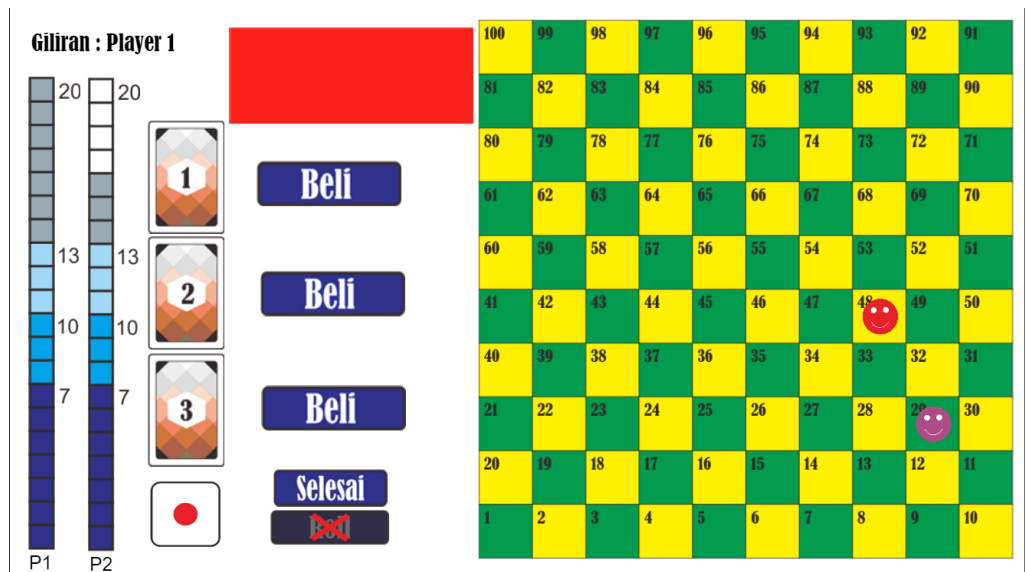
Player 2 Bergerak

Player2	Is move	Add action
System	Every 0.5 seconds	System Subtract 1 from JmlLangkahP2
System	JmlLangkahP2 > 0	Add action
Menambahkan poin P2		
System	PosisiP2x = 1	Player2 Set Sudah100 to True
System	PosisiP2y = 0	Add action
Player2	Is Sudah100	System Set DirectionP2 to "kanan"
		Add action
System	Else	System Set NilaiY to PosisiP2y%2
		Add action
Syst	NilaiY = 0	System Set DirectionP2 to "kiri"
		Add action
Syst	Else	System Set DirectionP2 to "kanan"
		Add action
System	DirectionP2 = "kiri"	Add action
Syst	PosisiP2x = 1	System Add -1 to PosisiP2y
		Add action
Syst	Else	System Add -1 to PosisiP2x
		Add action
System	DirectionP2 = "kanan"	Add action
Syst	PosisiP2x = 10	System Add -1 to PosisiP2y
		Add action
Syst	Else	System Add 1 to PosisiP2x
		Add action
Player2	GridMove move to [PosisiP2x, PosisiP2y]	
Audio	Play jump not looping at volume 0 dB (tag "jalan")	
		Add action
System	JmlLangkahP2 = 0	System Set faseGiliran to giliran
Dadu	Is Dadu2	Add action

Gambar 4.63 Source code event menggerakkan pion(pemain 2).



Gambar 4.64 Tampilan sebelum pion bergerak



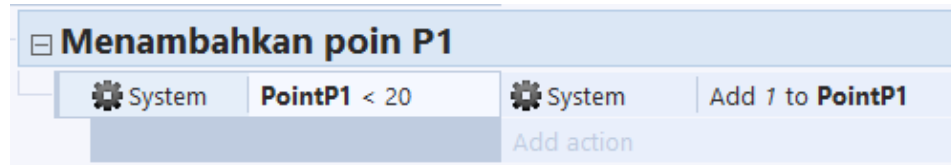
Gambar 4.65 Tampilan setelah pion bergerak

e. Event Menambahkan Poin

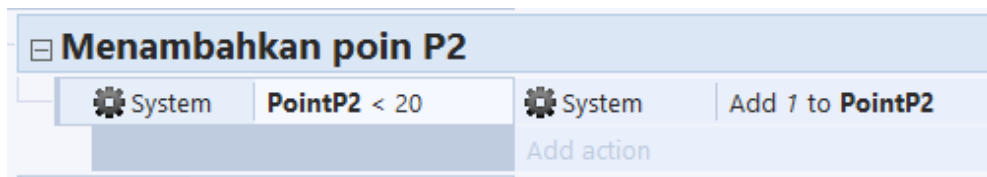
Event menambahkan poin ini berada pada *turn phase*, event ini dijalankan setelah event roll dadu dan bersamaan pada event menjalankan pion. Event ini dibuat berdasarkan pada sub bab 4.2.2 bagian b point 8. Event ini juga berfungsi untuk menampilkan poin setiap pemain.

Condition Commands yang digunakan pada event ini adalah *Compare variable*. *Action commands* pada event ini adalah *add value* dan *set animation*

frame. Dan ada 2 *variable* yang digunakan pada event ini yaitu PoinP1 dan PoinP2.



Gambar 4.66 *Source code* menambahkan poin (pemain 1).



Gambar 4.67 *Source code* menambahkan poin (pemain 2).

Event ini berjalan bersamaan dengan pion berjalan, ketika pion berjalan 1 langkah, sistem akan memeriksa *variable* poin, ketika poin kurang dari 20 maka sistem akan menambahkan *variable* poin senilai 1 angka. Dan ketika poin sama dengan 20, maka sistem tidak akan menambah nilai *variable* poin seperti yang dijelaskan pada gambar 4.66 dan 4.67. Poin di tampilkan setiap pion berjalan.

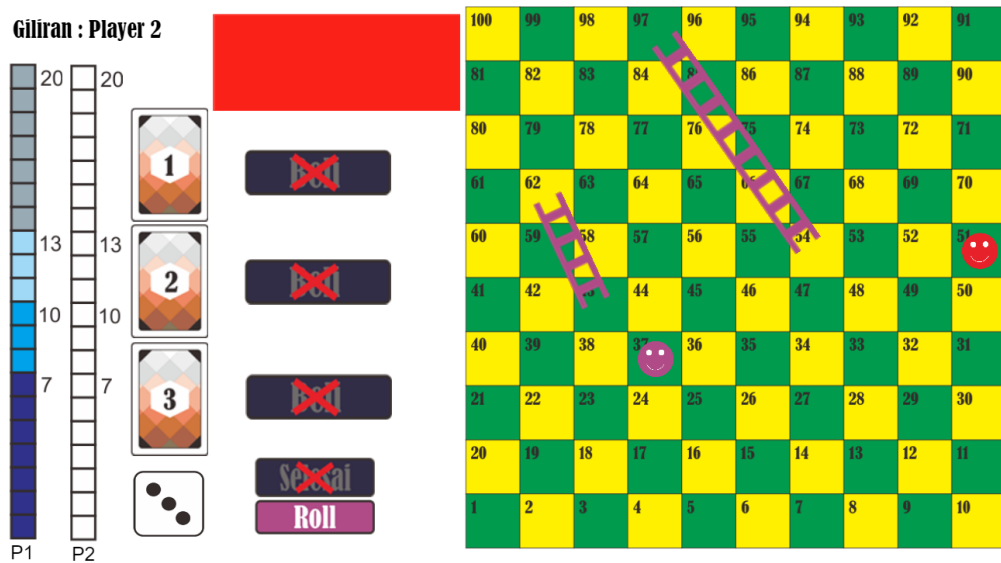
Source code event menampilkan dapat dilihat pada gambar 4.68 dan 4.69. Dan tampilan event menambahkan poin dapat dilihat pada gambar 4.70 dan 4.71.

PointP1			
System	PointP1 = 0	PointBarP1	Set animation frame to 0
		Add action	
System	PointP1 = 1	PointBarP1	Set animation frame to 1
		Add action	
System	PointP1 = 2	PointBarP1	Set animation frame to 2
		Add action	
System	PointP1 = 3	PointBarP1	Set animation frame to 3
		Add action	
System	PointP1 = 4	PointBarP1	Set animation frame to 4
		Add action	
System	PointP1 = 5	PointBarP1	Set animation frame to 5
		Add action	
System	PointP1 = 6	PointBarP1	Set animation frame to 6
		Add action	
System	PointP1 = 7	PointBarP1	Set animation frame to 7
		Add action	
System	PointP1 = 8	PointBarP1	Set animation frame to 8
		Add action	
System	PointP1 = 9	PointBarP1	Set animation frame to 9
		Add action	
System	PointP1 = 10	PointBarP1	Set animation frame to 10
		Add action	
System	PointP1 = 11	PointBarP1	Set animation frame to 11
		Add action	
System	PointP1 = 12	PointBarP1	Set animation frame to 12
		Add action	
System	PointP1 = 13	PointBarP1	Set animation frame to 13
		Add action	
System	PointP1 = 14	PointBarP1	Set animation frame to 14
		Add action	

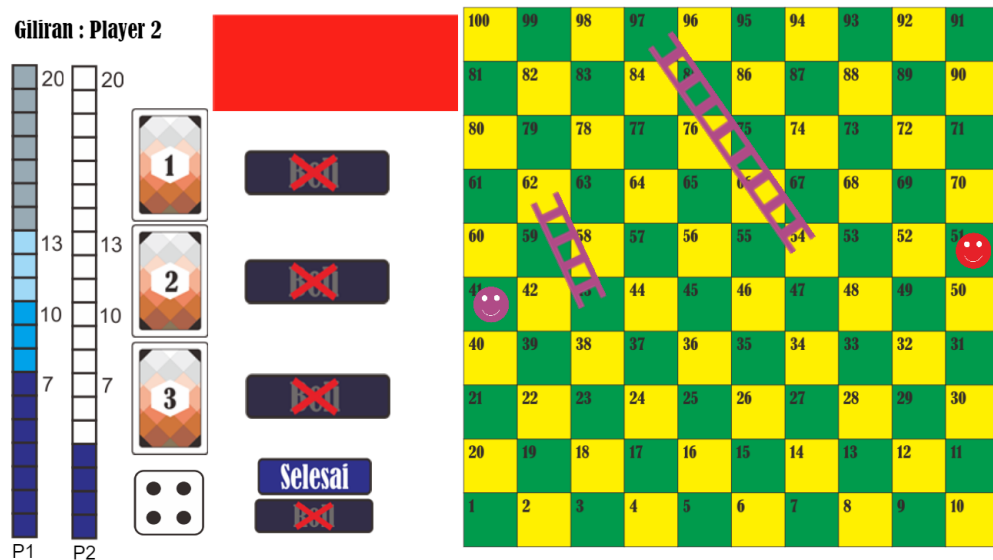
Gambar 4.68 Source code menampilkan poin (pemain 1).

PointP2			
System	PointP2 = 0	PointBarP2	Set animation frame to 0
Add action			
System	PointP2 = 1	PointBarP2	Set animation frame to 1
Add action			
System	PointP2 = 2	PointBarP2	Set animation frame to 2
Add action			
System	PointP2 = 3	PointBarP2	Set animation frame to 3
Add action			
System	PointP2 = 4	PointBarP2	Set animation frame to 4
Add action			
System	PointP2 = 5	PointBarP2	Set animation frame to 5
Add action			
System	PointP2 = 6	PointBarP2	Set animation frame to 6
Add action			
System	PointP2 = 7	PointBarP2	Set animation frame to 7
Add action			
System	PointP2 = 8	PointBarP2	Set animation frame to 8
Add action			
System	PointP2 = 9	PointBarP2	Set animation frame to 9
Add action			
System	PointP2 = 10	PointBarP2	Set animation frame to 10
Add action			
System	PointP2 = 11	PointBarP2	Set animation frame to 11
Add action			
System	PointP2 = 12	PointBarP2	Set animation frame to 12
Add action			
System	PointP2 = 13	PointBarP2	Set animation frame to 13
Add action			
System	PointP2 = 14	PointBarP2	Set animation frame to 14
Add action			

Gambar 4.69 *Source code* menampilkan poin (pemain 2).



Gambar 4.70 Tampilan sebelum poin bertambah



Gambar 4.71 Tampilan setelah poin bertambah

f. Event Membeli Kartu

Event membeli kartu ini berada pada *strategy phase*, event ini dijalankan setelah event-event yang ada di *turn phase* selesai dijalankan. Event ini dibuat berdasarkan pada sub bab 4.2.2 bagian b point 9. Event ini juga berfungsi untuk menentukan kartu yang didapat oleh pemain yang membelinya.

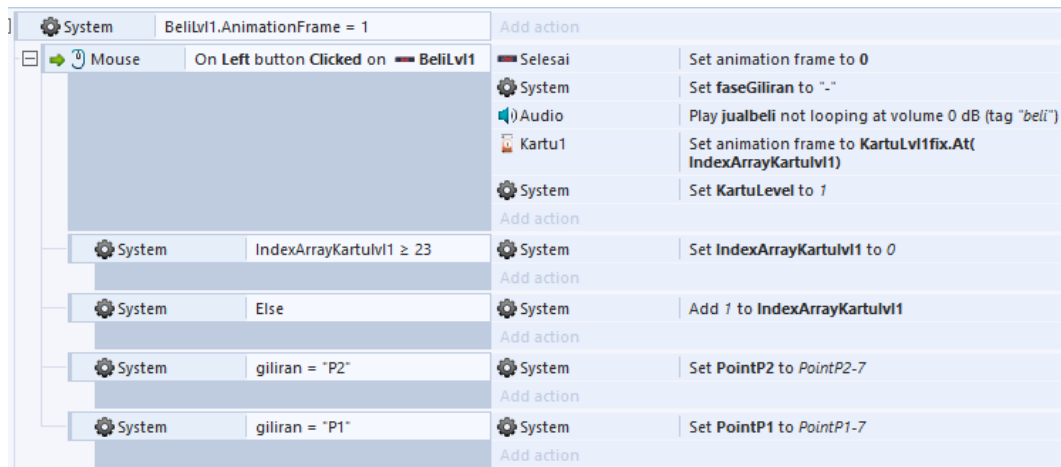
Condition Commands yang digunakan pada event ini adalah *Compare variable*, *Compare two values*, *Clicked*, dan *else*. *Action commands* pada event

ini adalah *set animation frame*, *set value*, *play*, *add value*, dan *set visible*. Dan ada 8 *variable* yang digunakan pada event ini yaitu PoinP1, PoinP2, FaseGiliran, KartuLevel, Giliran, *index* array dari setiap level kartu, Ular, dan Tangga.

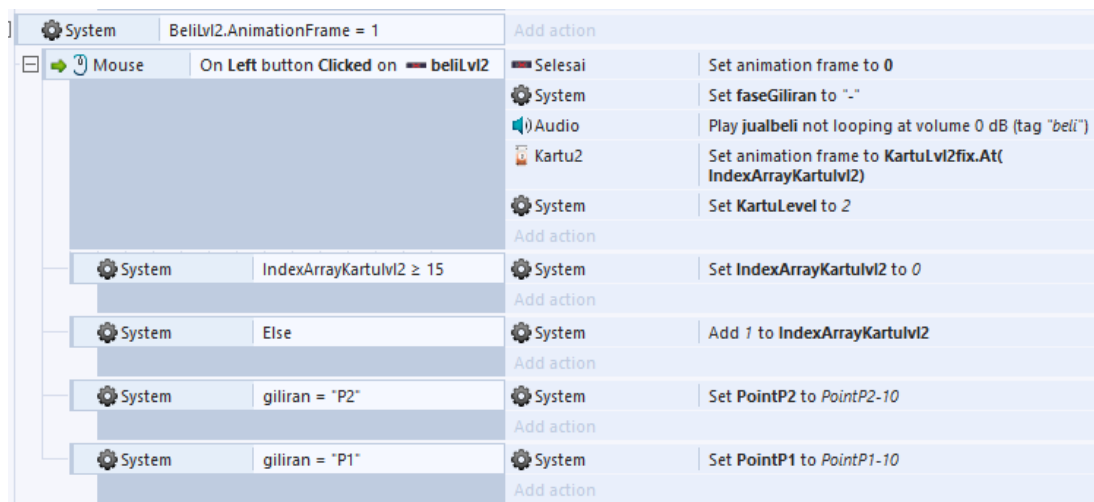
System	faseGiliran = "P1"	Add action
System	PointP1 ≥ 7	BeliLv1 Set animation frame to 1
		Add action
System	PointP1 < 7	BeliLv1 Set animation frame to 0
		Add action
System	PointP1 ≥ 10	beliLv2 Set animation frame to 1
		Add action
System	PointP1 < 10	beliLv2 Set animation frame to 0
		Add action
System	PointP1 ≥ 13	beliLv3 Set animation frame to 1
		Add action
System	PointP1 < 13	beliLv3 Set animation frame to 0
		Add action
System	faseGiliran = "P2"	Add action
System	PointP2 ≥ 7	BeliLv1 Set animation frame to 1
		Add action
System	PointP2 < 7	BeliLv1 Set animation frame to 0
		Add action
System	PointP2 ≥ 10	beliLv2 Set animation frame to 1
		Add action
System	PointP2 < 10	beliLv2 Set animation frame to 0
		Add action
System	PointP2 ≥ 13	beliLv3 Set animation frame to 1
		Add action
System	PointP2 < 13	beliLv3 Set animation frame to 0
		Add action

Gambar 4.72 Source code Mengaktifkan tombol beli

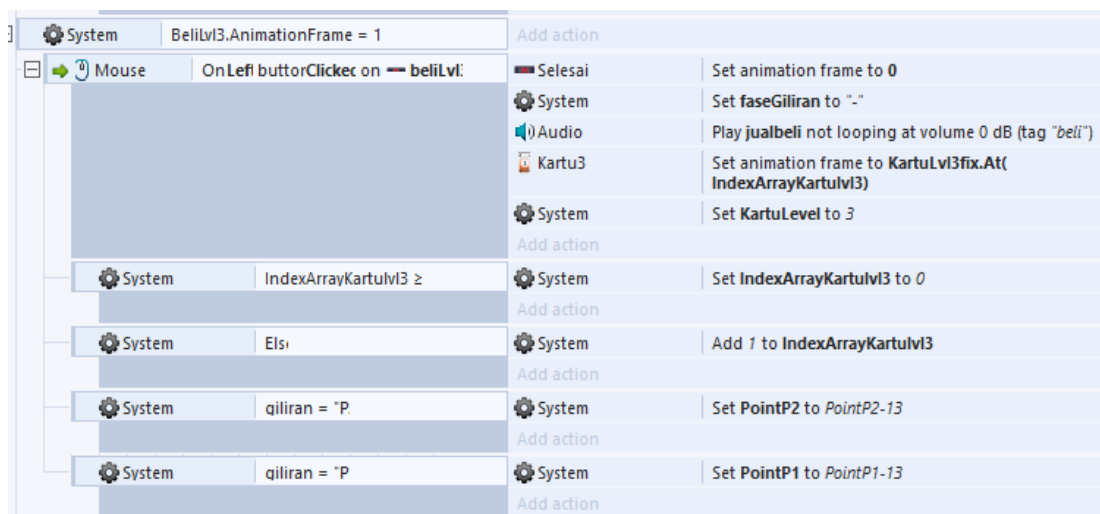
Setelah *turn phase* pemain selesai, sistem akan menjalankan *source code* pada gambar 4.72, pada gambar 4.72 dijelaskan sistem akan memeriksa jumlah poin pemain sesuai dengan ketentuan. Setelah diperiksa, sistem akan mengaktifkan tombol beli yang bisa digunakan sesuai jumlah poin yang dimiliki.



Gambar 4.73 Source code membeli kartu level 1



Gambar 4.74 Source code membeli kartu level 2



Gambar 4.75 Source code membeli kartu level 3

Pada gambar 4.73, 4.74, dan 4.75 menjelaskan *source code* setelah tombol beli aktif, dan pemain menekan tombol beli baik level 1, level 2, maupun level 3. Setelah pemain menekan tombol beli, sistem akan langsung menampilkan *frame* kartu berdasarkan urutan yang telah diacak pada event acak kartu. Setelah *frame* kartu berdasarkan urutan yang telah diacak sudah muncul sistem akan mengurangi poin pemain berdasarkan kartu yang dibeli.

Keluaran Animasi Kartu lvl1			
System	Kartu1.AnimationFrame = 0	TempatKartuLevel1	Set animation frame to 0
		TempatKartuLevel1	Set Invisible
Add action			
System	Kartu1.AnimationFrame = 1	System	Set Ular to "lvl1Ke1"
		TempatKartuLevel1	Set animation frame to 1
		TempatKartuLevel1	Set Visible
		TempatBKSetelahBeli	Set animation frame to 1
Add action			
System	Kartu1.AnimationFrame = 2	System	Set Ular to "lvl1Ke2"
		TempatKartuLevel1	Set animation frame to 1
		TempatKartuLevel1	Set Visible
		TempatBKSetelahBeli	Set animation frame to 1
Add action			
System	Kartu1.AnimationFrame = 3	System	Set Ular to "lvl1Ke3"
		TempatKartuLevel1	Set animation frame to 2
		TempatKartuLevel1	Set Visible
		TempatBKSetelahBeli	Set animation frame to 1
Add action			
System	Kartu1.AnimationFrame = 4	System	Set Ular to "lvl1Ke4"
		TempatKartuLevel1	Set animation frame to 2
		TempatKartuLevel1	Set Visible
		TempatBKSetelahBeli	Set animation frame to 1
Add action			
System	Kartu1.AnimationFrame = 5	TempatKartuLevel1	Set animation frame to 3
		TempatKartuLevel1	Set Visible
		System	Set Ular to "lvl1Ke5"
		TempatBKSetelahBeli	Set animation frame to 1
Add action			
System	Kartu1.AnimationFrame = 6	TempatKartuLevel1	Set animation frame to 3
		TempatKartuLevel1	Set Visible

Gambar 4.76 *Source code* memperbesar kartu Level 1

Keluaran Animasi Kartu lvl2			
System	Kartu2.AnimationFrame = 0	TempatKartuLevel2	Set animation frame to 0
		TempatKartuLevel2	Set Invisible
Add action			
System	Kartu2.AnimationFrame = 1	System	Set Ular to "lvl2Ke1"
		TempatKartuLevel2	Set animation frame to 1
		TempatKartuLevel2	Set Visible
		TempatBKSetelahBeli	Set animation frame to 1
Add action			
System	Kartu2.AnimationFrame = 2	System	Set Ular to "lvl2Ke2"
		TempatKartuLevel2	Set animation frame to 1
		TempatKartuLevel2	Set Visible
		TempatBKSetelahBeli	Set animation frame to 1
Add action			
System	Kartu2.AnimationFrame = 3	System	Set Ular to "lvl2Ke3"
		TempatKartuLevel2	Set animation frame to 2
		TempatKartuLevel2	Set Visible
		TempatBKSetelahBeli	Set animation frame to 1
Add action			
System	Kartu2.AnimationFrame = 4	System	Set Ular to "lvl2Ke4"
		TempatKartuLevel2	Set animation frame to 2
		TempatKartuLevel2	Set Visible
		TempatBKSetelahBeli	Set animation frame to 1
Add action			
System	Kartu2.AnimationFrame = 5	System	Set Ular to "lvl2Ke5"
		TempatKartuLevel2	Set animation frame to 3
		TempatKartuLevel2	Set Visible
		TempatBKSetelahBeli	Set animation frame to 1
Add action			
System	Kartu2.AnimationFrame = 6	System	Set Ular to "lvl2Ke6"
		TempatKartuLevel2	Set animation frame to 3
		TempatKartuLevel2	Set Visible

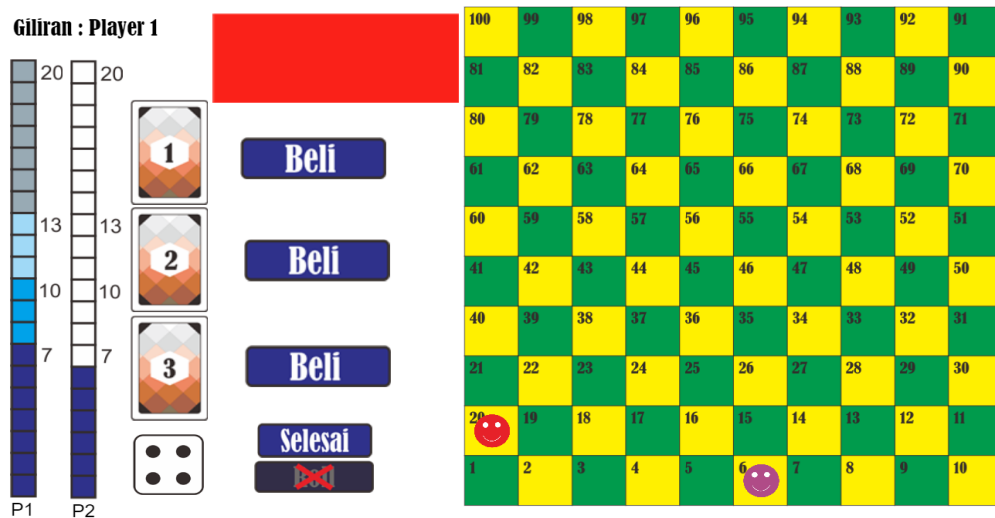
Gambar 4.77 Source code memperbesar kartu Level 2

Keluaran Animasi Kartu lv3		
System	Kartu3.AnimationFrame = 0	TempatKartuLevel3 Set animation frame to 0 TempatKartuLevel3 Set Invisible Add action
System	Kartu3.AnimationFrame = 1	System Set Ular to "lv3Ke1" TempatKartuLevel3 Set animation frame to 1 TempatKartuLevel3 Set Visible TempatBKSetelahBeli Set animation frame to 1 Add action
System	Kartu3.AnimationFrame = 2	System Set Ular to "lv3Ke2" TempatKartuLevel3 Set animation frame to 2 TempatKartuLevel3 Set Visible TempatBKSetelahBeli Set animation frame to 1 Add action
System	Kartu3.AnimationFrame = 3	System Set Ular to "lv3Ke3" TempatKartuLevel3 Set animation frame to 3 TempatKartuLevel3 Set Visible TempatBKSetelahBeli Set animation frame to 1 Add action
System	Kartu3.AnimationFrame = 4	System Set Ular to "lv3Ke4" TempatKartuLevel3 Set animation frame to 4 TempatKartuLevel3 Set Visible TempatBKSetelahBeli Set animation frame to 1 Add action
System	Kartu3.AnimationFrame = 5	System Set Ular to "lv3Ke5" TempatKartuLevel3 Set animation frame to 5 TempatKartuLevel3 Set Visible TempatBKSetelahBeli Set animation frame to 1 Add action
System	Kartu3.AnimationFrame = 6	System Set Ular to "lv3Ke6" TempatKartuLevel3 Set animation frame to 6 TempatKartuLevel3 Set Visible

Gambar 4.78 *Source code* memperbesar kartu Level 3

Gambar 4.76, 4.77, dan 4.78 menampilkan *source code* memperbesar kartu/menampilkan pada *card on hand*. Gambar 4.76, 4.77, dan 4.78 menjelaskan setelah poin berkurang sistem akan memperbesar kartu/menampilkan pada *card on hand* baik level 1, level 2, maupun level 3, agar kartu yang muncul dapat terlihat dan terbaca dengan jelas. Setelah kartu diperbesar, maka sistem akan menampilkan tombol beli dan tombol jual dan sistem akan set *variable* ular/tangga sesuai kartu yang didapat untuk digunakan pada event pasang ular atau tangga.

Tampilan event membeli kartu dapat dilihat pada gambar 4.79 dan 4.80.



Gambar 4.79 Tampilan sebelum membeli kartu



Gambar 4.80 Tampilan setelah membeli kartu

g. Event Jual Kartu

Event jual kartu ini berada pada *strategy phase*, event ini dijalankan setelah event membeli kartu dan ketika pemain ingin menjual kartu yang didapat. Event ini dibuat berdasarkan pada sub bab 4.2.2 bagian b point 9. Event ini juga berfungsi untuk menjual kartu yang didapat karena tidak sesuai yang diharapkan pemain.

Condition Commands yang digunakan pada event ini adalah *Clicked*, *is visible*, *Compare Variable*, dan *else*. *Action commands* pada event ini adalah

set animation frame, set value, play, add value, dan set visible. Dan ada 4 variable yang digunakan pada event ini yaitu PoinP1, PoinP2, FaseGiliran, dan KartuLevel.

Event	Condition	Action
Mouse	On Left button Clicked on BtnJual	TempatBKSetelahBeli: Set animation frame to 0
BtnJual	Is visible	TempatBKSetelahBeli: Set Invisible
System	KartuLevel = 1	BtnJual: Set Invisible
System	giliran = "P1"	BtnPasang: Set Invisible
System	Else	TextJualBeli: Set Invisible
System	KartuLevel = 2	Audio: Play jualbeli not looping at volume 0 dB (tag "jual")
System	giliran = "P1"	Add action
System	Else	Kartu1: Set animation frame to 0
System	KartuLevel = 3	System: Add 4 to PointP1
System	giliran = "P1"	System: Set KartuLevel to 0
System	Else	Selesai: Set animation frame to 1
System	KartuLevel = 1	System: Set faseGiliran to giliran
System	giliran = "P2"	Add action
System	Else	Kartu2: Set animation frame to 0
System	KartuLevel = 2	System: Add 6 to PointP1
System	giliran = "P2"	System: Set KartuLevel to 0
System	Else	Selesai: Set animation frame to 1
System	KartuLevel = 3	System: Set faseGiliran to giliran
System	giliran = "P2"	Add action
System	Else	Kartu3: Set animation frame to 0
System	KartuLevel = 1	System: Add 8 to PointP1
System	giliran = "P2"	System: Set KartuLevel to 0
System	Else	Selesai: Set animation frame to 1
System	KartuLevel = 2	System: Set faseGiliran to giliran
System	giliran = "P2"	Add action
System	Else	Kartu1: Set animation frame to 0
System	KartuLevel = 3	System: Add 4 to PointP2
System	giliran = "P2"	System: Set KartuLevel to 0
System	Else	Selesai: Set animation frame to 1
System	KartuLevel = 1	System: Set faseGiliran to giliran
System	giliran = "P2"	Add action
System	Else	Kartu2: Set animation frame to 0
System	KartuLevel = 2	System: Add 6 to PointP2
System	giliran = "P2"	System: Set KartuLevel to 0
System	Else	Selesai: Set animation frame to 1
System	KartuLevel = 3	System: Set faseGiliran to giliran
System	giliran = "P2"	Add action
System	Else	Kartu3: Set animation frame to 0
System	KartuLevel = 1	System: Add 8 to PointP2
System	giliran = "P2"	System: Set KartuLevel to 0
System	Else	Selesai: Set animation frame to 1
System	KartuLevel = 2	System: Set faseGiliran to giliran
System	giliran = "P2"	Add action
System	Else	Kartu3: Set animation frame to 0
System	KartuLevel = 3	System: Add 8 to PointP2
System	giliran = "P2"	System: Set KartuLevel to 0
System	Else	Selesai: Set animation frame to 1
System	KartuLevel = 1	System: Set faseGiliran to giliran
System	giliran = "P2"	Add action

Gambar 4.81 Source code menjual kartu

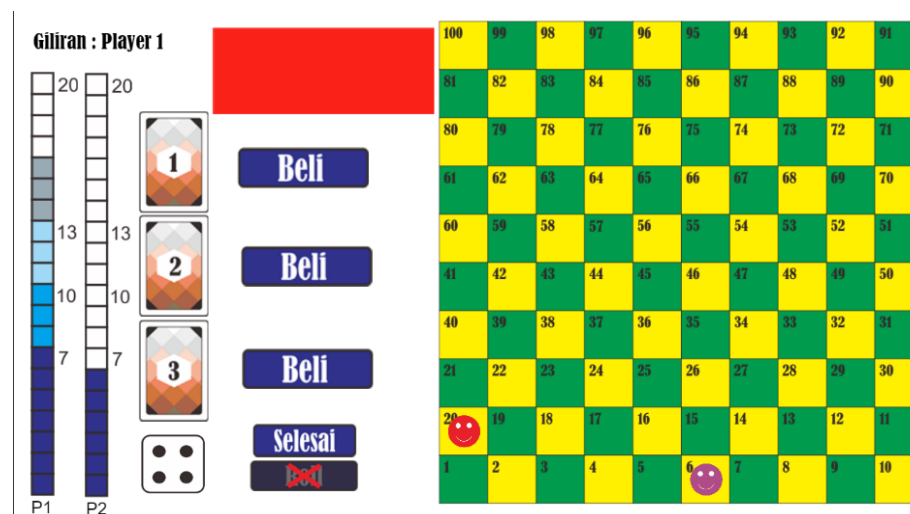
Pada gambar 4.81 menampilkan *source code* jual kartu. gambar 4.81 menjelaskan setelah pemain melakukan pembelian kartu, sistem akan

menampilkan 2 tombol yaitu tombol jual dan tombol pasang. Ketika pemain melakukan penjualan kartu dan menekan tombol jual, sistem *set visible* tombol jual dan tombol pasang menjadi *invisible* agar tombol tidak bisa digunakan, sistem akan *set animation frame* kartu menjadi 0 agar animasi kartu kembali seperti sebelum dibeli, sistem menambahkan poin ke *variable* poin pemain sesuai harga jual kartu. Dan tombol selesai diaktifkan untuk pemain bisa mengakhiri gilirannya, dan sistem mengembalikan *variable* faseGiliran ke giliran, agar ketika poin mencukupi pemain bisa membeli ulang kartunya.

Tampilan event jual kartu dapat dilihat pada gambar 4.82 dan 4.83.



Gambar 4.82 Tampilan sebelum jual kartu

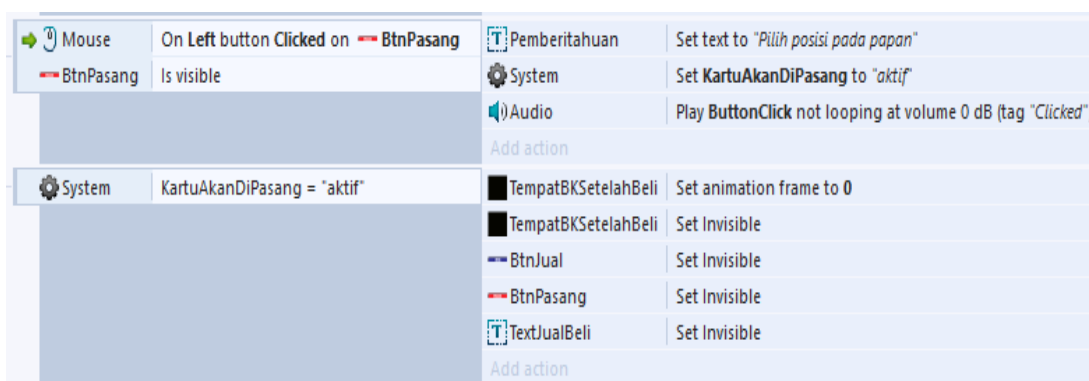


Gambar 4.83 Tampilan setelah jual kartu

h. Event Pemasangan Objek Ular atau Tangga

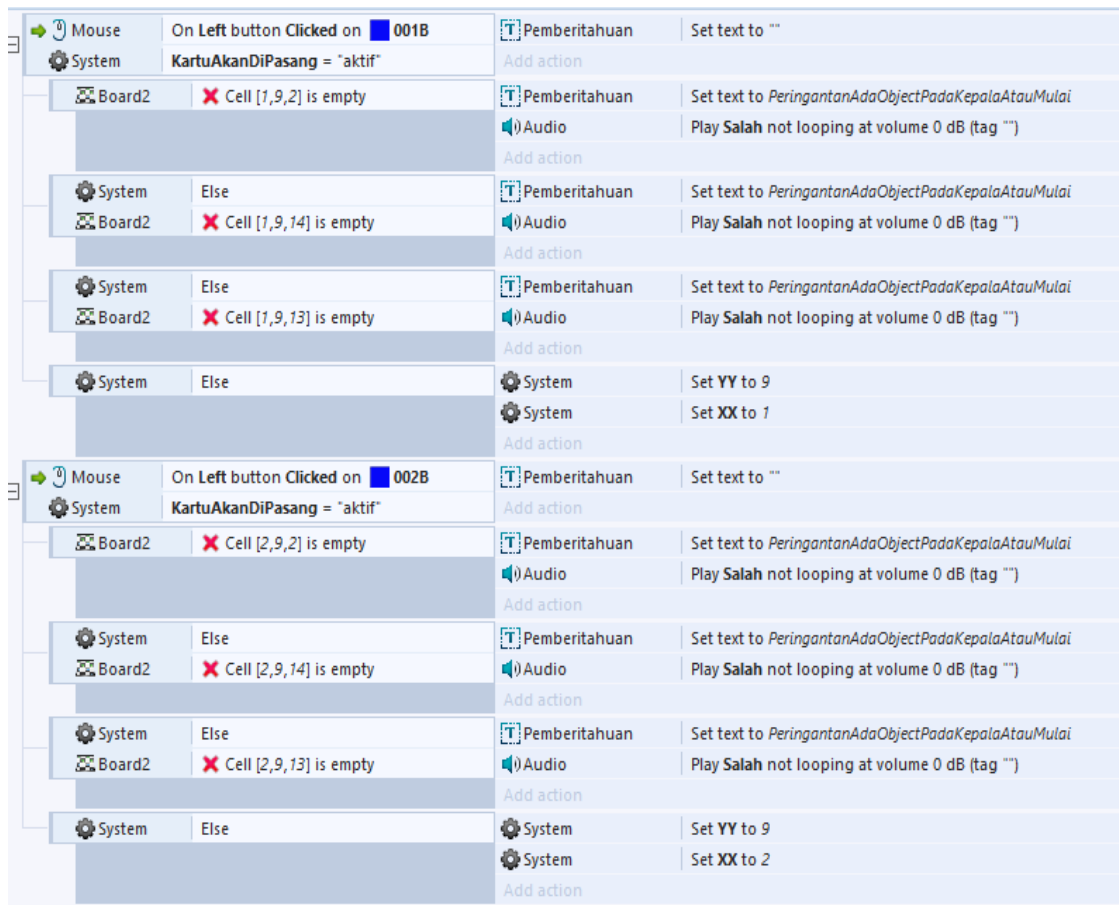
Event Pemasangan object ular atau tangga ini berada pada *strategy phase*, event ini dijalankan setelah event membeli kartu dan ketika pemain ingin memasang kartu yang didapat. Event ini dibuat berdasarkan pada sub bab 4.2.2 bagian b point 9. Event ini juga berfungsi untuk memasang kartu yang didapat pada papan permainan.

Condition Commands yang digunakan pada event ini adalah *Clicked*, *is visible*, *Compare two values*, *Compare Variable*, *is Empty*, dan *else*. *Action commands* pada event ini adalah *set text*, *set animation frame*, *set value*, *play*, *create chess*, *create object*, *wait*, dan *set visible*. Dan ada 4 *variable* yang digunakan pada event ini yaitu *KartuAkanDiPasang*, *umur*, *pemilik*, *XX*, *YY*, *PosisiEkorXX*, *PosisiEkorYY*, *PosisiUjungXX*, *PosisiUjungYY*, *PosisiMulaiSetelahTurunX*, *PosisiMulaiSetelahTurunY*, *PosisiMulaiSetelahNaikX*, *PosisiMulaiSetelahNaikY*, *Ular*, dan *Tangga*.

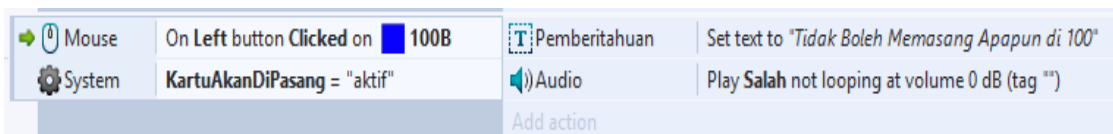


Gambar 4.84 Source code ketika pemain menekan tombol pasang.

Pada gambar 4.84 menampilkan *source code* ketika pemain menekan tombol pasang. Setelah pemain melakukan pembelian kartu, sistem akan menampilkan 2 tombol yaitu tombol jual dan tombol pasang. Gambar 4.47 menjelaskan ketika pemain ingin memasang objek yang ada di kartu dan menekan tombol pasang. Setelah tombol pasang ditekan, sistem menampilkan *text* pilih posisi pada papan dan set value pada *variable* *KartuAkanDiPasang* menjadi aktif untuk membuat papan bisa dipasang objek.



Gambar 4.85 Salah satu *source code* pemilihan kotak dan pemeriksaan kotak yang dipilih serta menentukan nilai *variable* XX dan YY



Gambar 4.86 *Source code* ketika kotak yang dipilih adalah kotak 100

Pada gambar 4.85 dan 4.86 menjelaskan *sourcecode* ketika *variable* *KartuAkanDiPasang* menjadi aktif, maka pemain diharuskan memilih kotak pada papan untuk posisi kepala ular atau awal tangga. Ketika pemain sudah memilih kotak pada papan, sistem akan memeriksa kotak yang dipilih. Saat kotak yang dipilih ada objek lain dan kotak yang dipilih adalah kotak 100 maka sistem akan memunculkan peringatan, ketika kotak yang dipilih tidak ada objek dan bukan kotak 100, sistem akan set *variable* XX dan YY menjadi nilai x dan y kotak yang dipilih.

System	XX = -1	System	Set PosisiEkorXX to XX
System	YY = -1	System	Set PosisiEkorYY to YY+1
System	Ular = "l1Ke1"	Add action	
System	PosisiEkorXX ≤ 10	Add action	
System	PosisiEkorYY ≤ 9		
System	PosisiEkorXX ≥ 1		
System	PosisiEkorYY ≥ 0		
System	Else	Pemberitahuan	Set text to PeringatanKeluarArena
		System	Set XX to -1
		System	Set YY to -1
		Audio	Play Salah not looping at volume 0 dB (tag "")
		Add action	

Gambar 4.87 Salah satu *source code* set *variable* PosisiEkorXX dan PosisiEkorYY serta pemeriksaan apakah posisi ekor ular keluar papan atau tidak.

System	XX = -1	System	Set PosisiUjungXX to XX
System	YY = -1	System	Set PosisiUjungYY to YY-1
System	Tangga = "l1Ke1"	Add action	
System	PosisiUjungXX ≤ 10	Add action	
System	PosisiUjungYY ≤ 9		
System	PosisiUjungXX ≥ 1		
System	PosisiUjungYY ≥ 0		
System	PosisiUjungXX = 1	Pemberitahuan	Set text to "ujung tangga tidak boleh ke 100"
System	PosisiUjungYY = 0	System	Set XX to -1
		System	Set YY to -1
		Audio	Play Salah not looping at volume 0 dB (tag "")
		Add action	
System	Else	Add action	
System	Else	Pemberitahuan	Set text to PeringatanKeluarArena
		System	Set XX to -1
		System	Set YY to -1
		Audio	Play Salah not looping at volume 0 dB (tag "")
		Add action	

Gambar 4.88 Salah satu *source code* set *variable* PosisiUjungXX dan PosisiUjungYY, pemeriksaan apakah posisi Ujung tangga keluar papan atau tidak, serta pemeriksaan apakah ujung tangga mengarah ke kotak 100.

Gambar 4.87 dan 4.88 menjelaskan *source code* ketika *variable* XX dan YY sudah tidak sama dengan -1/sudah diisi dengan nilai x dan y kotak yang dipilih dan *variable* ular atau tangga sesuai dengan yang tertera dalam kartu, maka sistem langsung set PosisiEkorXX dan PosisiEkorYY atau PosisiUjungXX dan PosisiUjungYY dengan ketentuan sesuai yang di kartu. Dan ketika PosisiEkorXX dan PosisiEkorYY atau PosisiUjungXX dan PosisiUjungYY keluar papan permainan atau tepat di kotak 100 maka sistem

akan menampilkan peringatan dan mereset nilai XX dan YY serta pemain melakukan pemilihan ulang kotak pada papan dengan meilih kotak yang lain.

Board2	Cell [PosisiEkorXX, PosisiEkorYY,2] is empty	Pemberitahuan	Set text to <i>PeringatanAdaObjectPadaEkorAtauUjung</i>
		System	Set XX to -1
		System	Set YY to -1
		Audio	Play Salah not looping at volume 0 dB (tag "")
		Add action	
System	Else	Pemberitahuan	Set text to <i>PeringatanAdaObjectPadaEkorAtauUjung</i>
Board2	Cell [PosisiEkorXX, PosisiEkorYY,13] is empty	System	Set XX to -1
		System	Set YY to -1
		Audio	Play Salah not looping at volume 0 dB (tag "")
		Add action	
System	Else	Pemberitahuan	Set text to <i>PeringatanAdaObjectPadaEkorAtauUjung</i>
Board2	Cell [PosisiEkorXX, PosisiEkorYY,14] is empty	System	Set XX to -1
		System	Set YY to -1
		Audio	Play Salah not looping at volume 0 dB (tag "")
		Add action	

Gambar 4.89 Salah satu *source code* pemeriksaan apakah posisi ekor ular ada objek lain atau tidak.

Board2	Cell [PosisiUjungXX, PosisiUjungYY,2] is empty	Pemberitahuan	Set text to <i>PeringatanAdaObjectPadaEkorAtauUjung</i>
		System	Set XX to -1
		System	Set YY to -1
		Audio	Play Salah not looping at volume 0 dB (tag "")
		Add action	
System	Else	Pemberitahuan	Set text to <i>PeringatanAdaObjectPadaEkorAtauUjung</i>
Board2	Cell [PosisiUjungXX, PosisiUjungYY,13] is empty	System	Set XX to -1
		System	Set YY to -1
		Audio	Play Salah not looping at volume 0 dB (tag "")
		Add action	
System	Else	Pemberitahuan	Set text to <i>PeringatanAdaObjectPadaEkorAtauUjung</i>
Board2	Cell [PosisiUjungXX, PosisiUjungYY,14] is empty	System	Set XX to -1
		System	Set YY to -1
		Audio	Play Salah not looping at volume 0 dB (tag "")
		Add action	

Gambar 4.90 Salah satu *source code* pemeriksaan apakah posisi ekor ular ada objek lain atau tidak.

Gambar 4.89 dan 4.90 menjelaskan *source code* ketika PosisiEkorXX dan PosisiEkorYY atau PosisiUjungXX dan PosisiUjungYY ada objek lain, maka sistem akan menampilkan peringatan dan mereset nilai XX dan YY serta pemain melakukan pemilihan ulang kotak pada papan dengan meilih kotak yang lain.

System	Else	Kartu1	Set animation frame to 0
		Board2	Create chess UlarB1Y1 to [XX, YY, 2] on layer 1
		Board2	Create chess UlarB1Y1Finish to [PosisiEkorXX, PosisiEkorYY, 2] on layer 1
		UlarB1Y1	Set umur to <i>Umur</i>
		UlarB1Y1	Set pemilik to <i>giliran</i>
		UlarB1Y1Finish	Set PosisiMulaiSetelahTurunX to <i>PosisiEkorXX</i>
		UlarB1Y1Finish	Set PosisiMulaiSetelahTurunY to <i>PosisiEkorYY</i>
		System	Set XX to -1
		System	Set YY to -1
		System	Set KartuAkanDiPasang to "tidak aktif"
		System	Set Ular to "tidak aktif"
		Selesai	Set animation frame to 1
		Pemberitahuan	Set text to "benar"
		System	Set faseGiliran to <i>giliran</i>
		Audio	Play PasangUlar not looping at volume 0 dB (tag "PasangUlar")
		Add action	
System	UlarB1Y1.pemilik = "P1"	System	Create object UlarLv1Ke1 on layer 2 at (<i>UlarB1Y1.X</i> , <i>UlarB1Y1.Y+23</i>)
		UlarLv1Ke1	Set animation frame to 0
		System	Wait 1.5 seconds
		Pemberitahuan	Set text to ""
		Add action	
System	Else	System	Create object UlarLv1Ke1 on layer 2 at (<i>UlarB1Y1.X</i> , <i>UlarB1Y1.Y+23</i>)
System	UlarB1Y1.pemilik = "P2"	UlarLv1Ke1	Set animation frame to 1
		System	Wait 1.5 seconds
		Pemberitahuan	Set text to ""

Gambar 4.91 Salah satu *source code* pemasangan objek ular

System	Else	Kartu1	Set animation frame to 0
		Board2	Create chess ■ TanggaA1Y1 to [XX, YY, Z] on layer 1
		Board2	Create chess ■ TanggaA1Y1Finish to [PosisiUjungXX, PosisiUjungYY, Z] on layer 1
		TanggaA1Y1	Set umur to Umur
		TanggaA1Y1	Set pemilik to giliran
		TanggaA1Y1Finish	Set PosisiMulaiSetelahNaikX to PosisiUjungXX
		TanggaA1Y1Finish	Set PosisiMulaiSetelahNaikY to PosisiUjungYY
		System	Set XX to -1
		System	Set YY to -1
		System	Set KartuAkanDiPasang to "tidak aktif"
		System	Set Tangga to "tidak aktif"
		Selesai	Set animation frame to 1
		Pemberitahuan	Set text to "benar"
		System	Set faseGiliran to giliran
		Audio	Play Pasang Tangga not looping at volume 0 dB (tag "PasangTangga")
Add action			
System	TanggaA1Y1. pemilik = "P1"	System	Create object ■ Tanggalv1Ke1 on layer 2 at (TanggaA1Y1.X, TanggaA1Y1.Y-23)
		Tanggalv1Ke1	Set animation frame to 0
		System	Wait 1.5 seconds
		Pemberitahuan	Set text to ""
Add action			
System	Else	System	Create object ■ Tanggalv1Ke1 on layer 2 at (TanggaA1Y1.X, TanggaA1Y1.Y-23)
System	TanggaA1Y1. pemilik = "P2"	Tanggalv1Ke1	Set animation frame to 1
		System	Wait 1.5 seconds
		Pemberitahuan	Set text to ""

Gambar 4.92 Salah satu *source code* pemasangan objek tangga

Gambar 4.91 dan 4.92 menjelaskan *source code* ketika PosisiEkorXX dan PosisiEkorYY atau PosisiUjungXX dan PosisiUjungYY tidak ada objek, tidak keluar arena, dan tidak berada di kotak 100, sistem akan meletakkan posisi kepala ular dan ekor atau awal tangga dan ujung tangga pada nilai yang sudah ditentukan. Kepala ular dan awal tangga diletakan pada posisi XX dan YY, ekor diletakan pada PosisiEkorXX dan PosisiEkorYY, dan ujung tangga diletakan pada PosisiUjungXX dan PosisiUjungYY. Setelah itu sistem akan *set variable* umur, pemilik objek, PosisiMulaiSetelahTurunX dan PosisiMulaiSetelahTurunY atau PosisiMulaiSetelahNaikX dan

PosisiMulaiSetelahNaikY, dan sistem mengeluarkan animasi ular atau tangga pada tampilan papan permainan.

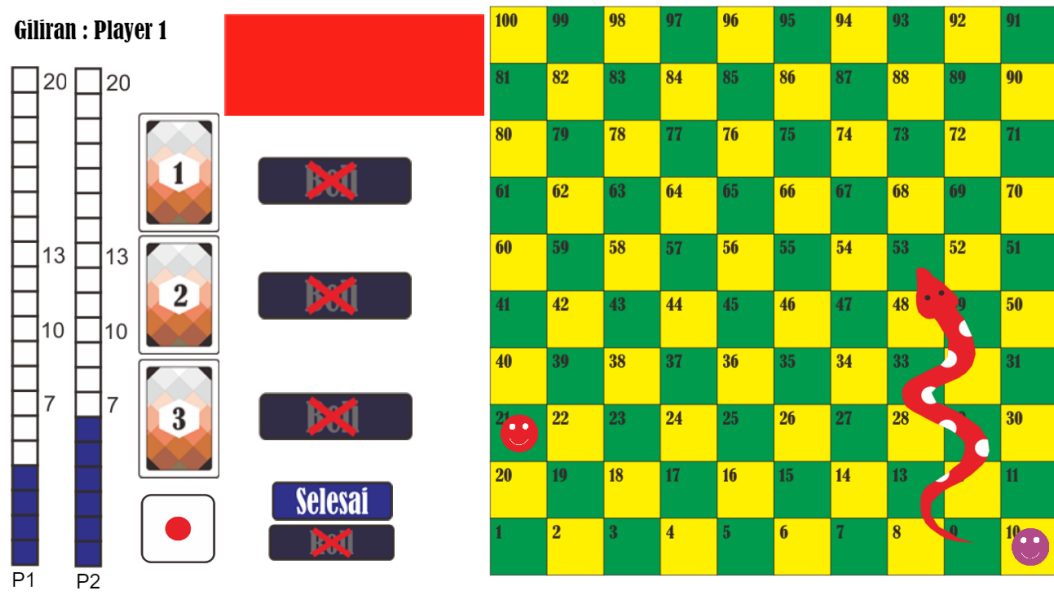
Tampilan event pemasangan objek ular atau tangga dapat dilihat pada gambar 4.93, 4.94, 4.95, 4.96, 4.97, dan 4.98.



Gambar 4.93 Tampilan sebelum pasang kartu Ular



Gambar 4.94 Tampilan memilih kotak untuk memasang kartu ular



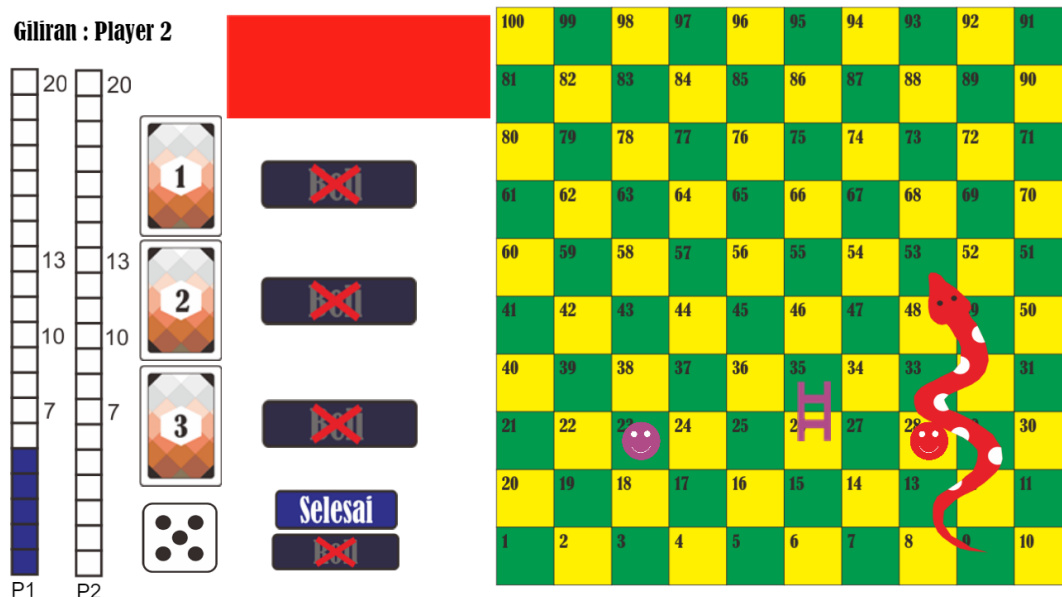
Gambar 4.95 Tampilan setelah memasang ular



Gambar 4.96 Tampilan sebelum pasang kartu Tangga



Gambar 4.97 Tampilan memilih kotak untuk memasang kartu tangga



Gambar 4.98 Tampilan setelah memasang tangga

i. Event Pion Mengenai Ular

Event Pion Mengenai Ular adalah event dimana pion berhenti pada kepala ular, event ini dijalankan setelah event menggerakkan pion. Event ini dibuat berdasarkan pada sub bab 4.2.2 bagian b point 10. Event ini juga berfungsi untuk menurunkan pion ketika pion berhenti di kepala ular lawan.

Condition Commands yang digunakan pada event ini adalah *Is Boolean*, *Is Overlapping*, dan *Compare two values*. *Action commands* pada event ini adalah *wait*, *play*, *gridMove*, *set value*, dan *Destroy*. Dan ada 5 *variable* yang digunakan pada event ini yaitu *Pemilik*, *PosisiMulaiSetelahTurunX*, *PosisiMulaiSetelahTurunY*, *PosisiP1x*, *PosisiP1y*, *PosisiP2x*, dan *PosisiP2y*.

Object	Condition	Action
Player1	Is move	System: Wait 0.5 seconds
Board2	Is UlarB1Y1 overlapping with Player1	Audio: Play turun not looping at volume 0 dB (tag "")
Player1	Is move	Player1: GridMove move to [UlarB1Y1Finish, PosisiMulaiSetelahTurunX, UlarB1Y1Finish, PosisiMulaiSetelahTurunY]
System	UlarB1Y1.pemilik = "P2"	System: Set PosisiP1x to UlarB1Y1Finish.PosisiMulaiSetelahTurunX
		System: Set PosisiP1y to UlarB1Y1Finish.PosisiMulaiSetelahTurunY
		System: Wait 0.5 seconds
UlarB1Y1		UlarB1Y1: Destroy
UlarB1Y1Finish		UlarB1Y1Finish: Destroy
UlarB1Y1		UlarB1Y1: Set pemilik to "-"
UlarLv1Ke1		UlarLv1Ke1: Destroy
Audio		Audio: Play UlarMati not looping at volume 0 dB (tag "UlarMati")
Board2	Is UlarB1Y2 overlapping with Player1	Audio: Play turun not looping at volume 0 dB (tag "")
Player1	Is move	Player1: GridMove move to [UlarB1Y2Finish, PosisiMulaiSetelahTurunX, UlarB1Y2Finish, PosisiMulaiSetelahTurunY]
System	UlarB1Y2.pemilik = "P2"	System: Set PosisiP1x to UlarB1Y2Finish.PosisiMulaiSetelahTurunX
		System: Set PosisiP1y to UlarB1Y2Finish.PosisiMulaiSetelahTurunY
		System: Wait 0.5 seconds
UlarB1Y2		UlarB1Y2: Destroy
UlarB1Y2Finish		UlarB1Y2Finish: Destroy
UlarB1Y2		UlarB1Y2: Set pemilik to "-"
UlarLv1Ke2		UlarLv1Ke2: Destroy
Audio		Audio: Play UlarMati not looping at volume 0 dB (tag "UlarMati")

Gambar 4.99 Salah satu *source code* pemain 1 terkena ular

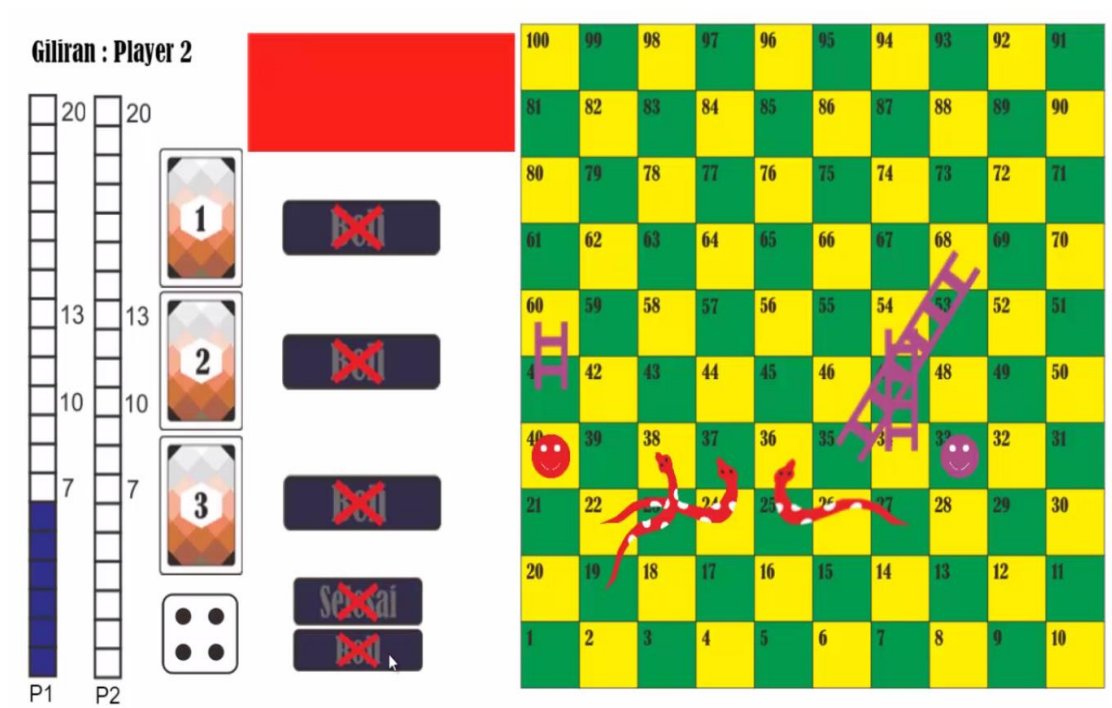


Gambar 4.100 Salah Satu *source code* pemain 2 terkena ular

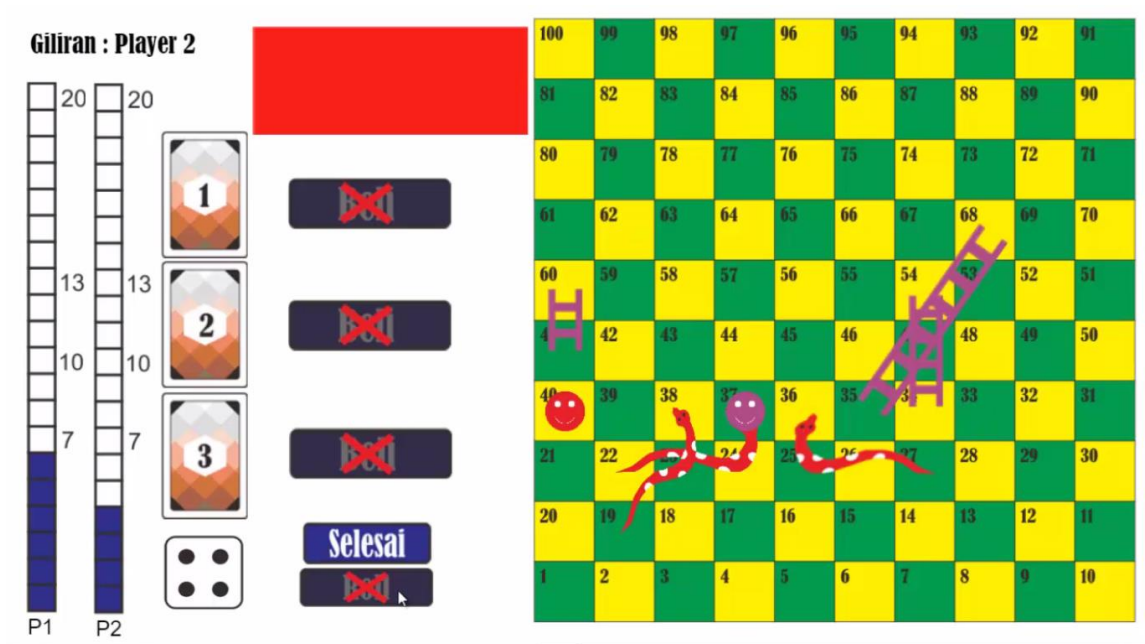
Gambar 4.99 dan 4.100 menjelaskan *source code* setelah pemain melakukan *turn phase* dan pion berhenti menginjak kepala ular. Setelah berhenti, sistem akan memeriksa apakah *boolean* move pemain bernilai *false*, ketika *boolean* move bernilai *false* maka sistem akan menunggu selama 0,5 detik untuk menjalankan langkah selanjutnya, setelah 0,5 detik sistem akan memeriksa, apakah pion berada pada kepala ular, apakah *boolean* move pemain bernilai *false*, dan apakah pemilik ular adalah lawan. Ketika salah satu pertanyaan tersebut atau semua pertanyaan tersebut tidak terpenuhi/bernilai tidak maka pion dianggap tidak menginjak ular dan pion tidak diturunkan/dipindahkan, namun ketika 3 pertanyaan tersebut terpenuhi/jawabannya bernilai iya maka pion di anggap menginjak ular dan pion

diturunkan/dipindahkan ke posisi yang sudah ditentukan. Setelah 3 pertanyaan tersebut terpenuhi/jawabanya bernilai iya, sistem akan menjalankan perintah *play* untuk membunyikan efek turun, setelah itu sistem akan memindahkan pion dengan perintah *gridMove* ke posisi nilai *variable* PosisiMulaiSetelahTurunX dan PosisiMulaiSetelahTurunY yang sudah di set ketika pemasangan ular. Dan *variable* PosisiP1x/ PosisiP2x dan PosisiP1y/ PosisiP2y juga di set menjadi nilai *variable* PosisiMulaiSetelahTurunX dan PosisiMulaiSetelahTurunY sebagai penanda bahwa pion berada di kotak yang berisi ekor ular. Setelah itu sistem akan menunggu selama 0.5 detik. Setelah 0,5 detik, sistem akan menghapus ekor, kepala, dan animasi ularnya dengan perintah *destroy*, dan *set variable* pemilik ular tersebut menjadi kosong, serta menjalankan efek suara ular mati.

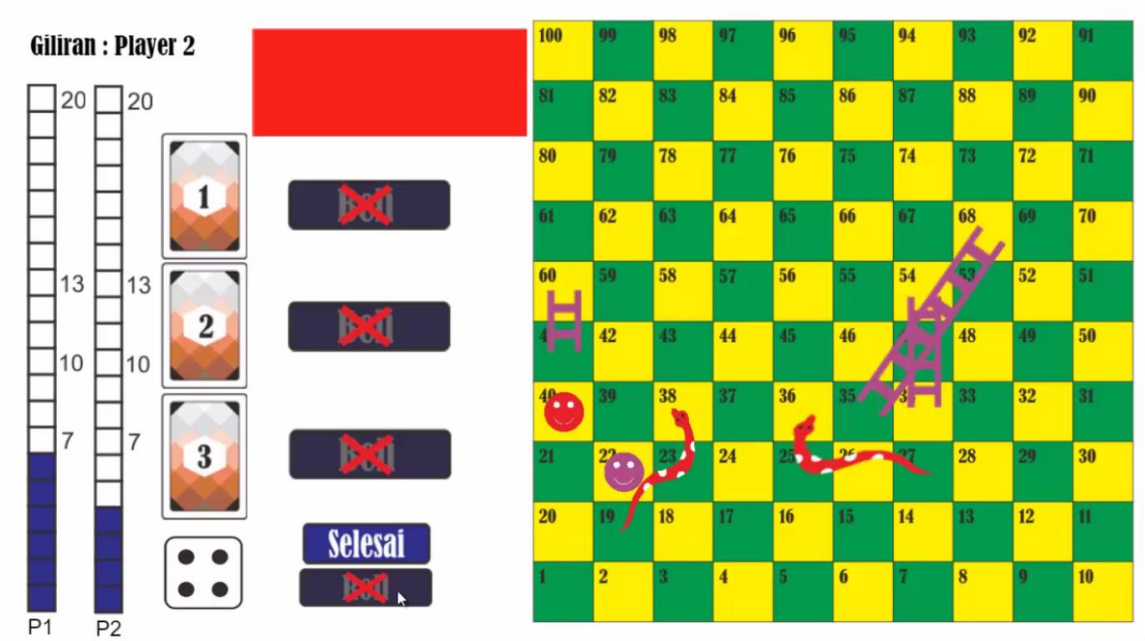
Tampilan event pion mengenai ular dapat dilihat pada gambar 4.101, 4.102, dan 4.103.



Gambar 4.101 Tampilan sebelum pion terkena ular



Gambar 4.102 Tampilan pion menginjak kepala ular



Gambar 4.103 Tampilan setelah pion turun akibat menginjak ular

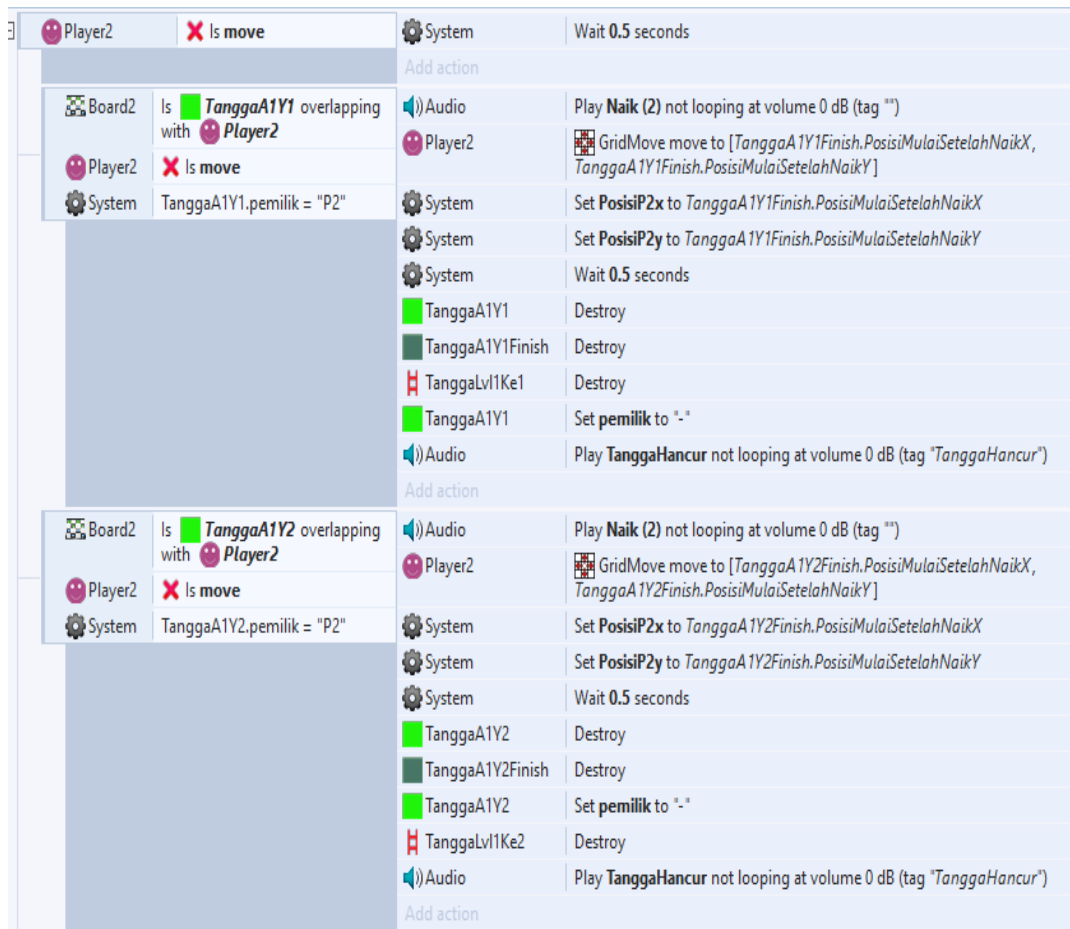
j. Event Pion Mengenai Tangga

Event Pion Mengenai Tangga adalah event dimana pion berhenti pada awalan tangga, event ini dijalankan setelah event menggerakkan pion. Event ini dibuat berdasarkan pada sub bab 4.2.2 bagian b point 10. Event ini juga berfungsi untuk menaikkan pion ketika pion berhenti di awalan tangga.

Condition Commands yang digunakan pada event ini adalah *Is Boolean*, *Is Overlapping*, dan *Compare two values*. *Action commands* pada event ini adalah *wait*, *play*, *gridMove*, *set value*, dan *Destroy*. Dan ada 5 *variable* yang digunakan pada event ini yaitu *Pemilik*, *PosisiMulaiSetelahNaikX*, *PosisiMulaiSetelahNaikY*, *PosisiP1x*, *PosisiP1y*, *PosisiP2x*, dan *PosisiP2y*.

Player1	Is move	System	Wait 0.5 seconds
Add action			
Board2	Is TanggaA1Y1 overlapping with Player1	Audio	Play Naik (2) not looping at volume 0 dB (tag "")
Player1	Is move	Player1	GridMove move to [<i>TanggaA1Y1Finish.PosisiMulaiSetelahNaikX</i> , <i>TanggaA1Y1Finish.PosisiMulaiSetelahNaikY</i>]
System	<i>TanggaA1Y1.pemilik = "P1"</i>	System	Set PosisiP1x to <i>TanggaA1Y1Finish.PosisiMulaiSetelahNaikX</i>
		System	Set PosisiP1y to <i>TanggaA1Y1Finish.PosisiMulaiSetelahNaikY</i>
		System	Wait 0.5 seconds
		TanggaA1Y1	Destroy
		TanggaA1Y1Finish	Destroy
		TanggaLv1Ke1	Destroy
		TanggaA1Y1	Set pemilik to "-"
		Audio	Play TanggaHancur not looping at volume 0 dB (tag " <i>TanggaHancur</i> ")
Add action			
Board2	Is TanggaA1Y2 overlapping with Player1	Audio	Play Naik (2) not looping at volume 0 dB (tag "")
Player1	Is move	Player1	GridMove move to [<i>TanggaA1Y2Finish.PosisiMulaiSetelahNaikX</i> , <i>TanggaA1Y2Finish.PosisiMulaiSetelahNaikY</i>]
System	<i>TanggaA1Y2.pemilik = "P1"</i>	System	Set PosisiP1x to <i>TanggaA1Y2Finish.PosisiMulaiSetelahNaikX</i>
		System	Set PosisiP1y to <i>TanggaA1Y2Finish.PosisiMulaiSetelahNaikY</i>
		System	Wait 0.5 seconds
		TanggaA1Y2	Destroy
		TanggaA1Y2Finish	Destroy
		TanggaA1Y2	Set pemilik to "-"
		TanggaLv1Ke2	Destroy
		Audio	Play TanggaHancur not looping at volume 0 dB (tag " <i>TanggaHancur</i> ")
Add action			

Gambar 4.104 Salah Satu *source code* pemain 1 terkena tangga

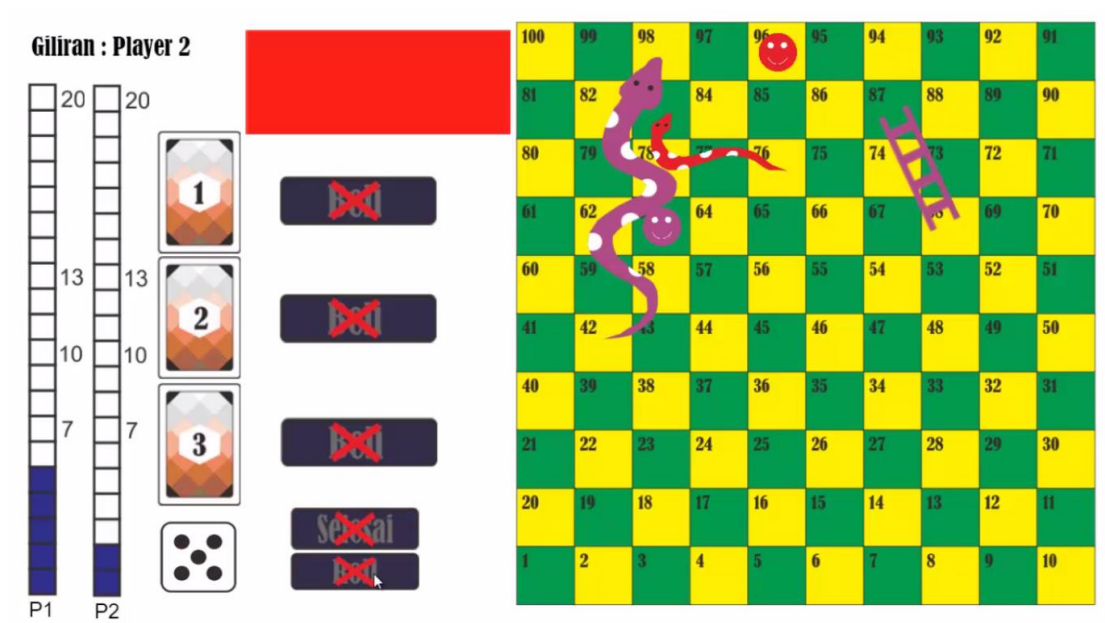


Gambar 4.105 Salah Satu *source code* pemain 2 terkena tangga

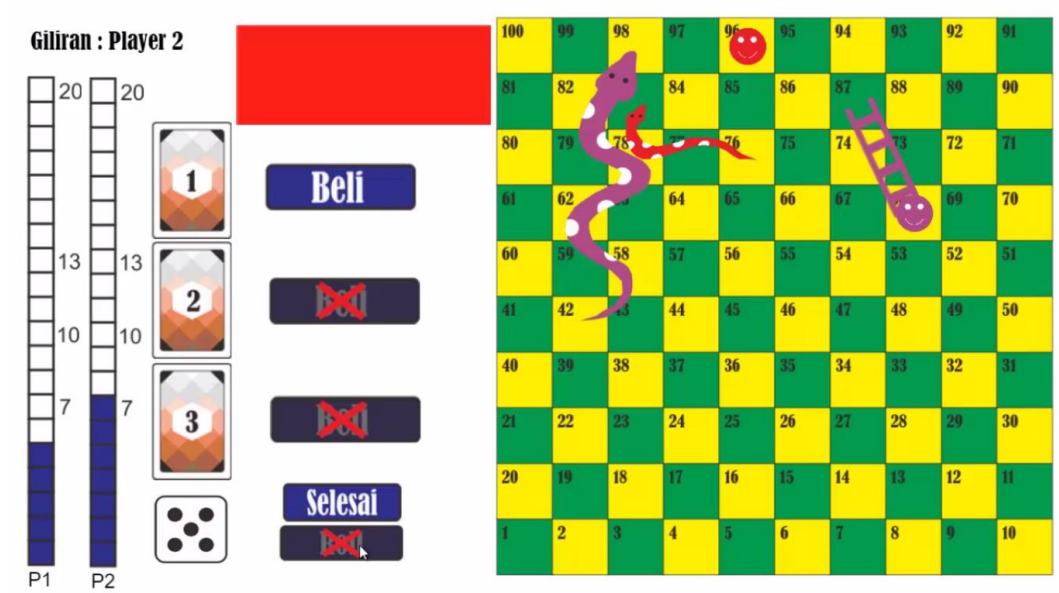
Gambar 4.104 dan 4.105 menjelaskan *source code* setelah pemain melakukan *turn phase* dan pion berhenti menginjak awalan tangga. Setelah pemain melakukan *turn phase* dan pion berhenti setelah berjalan/bergerak, sistem akan memeriksa apakah *boolean* move pemain bernilai *false*, ketika *boolean* move bernilai *false* maka sistem akan menunggu selama 0,5 detik untuk menjalankan langkah selanjutnya, setelah 0,5 detik sistem akan memeriksa, apakah pion berada pada awalan tangga, apakah *boolean* move pemain bernilai *false*, dan apakah pemilik ular adalah pemain itu sendiri. Ketika salah satu pertanyaan tersebut atau semua pertanyaan tersebut tidak terpenuhi/bernilai tidak maka pion dianggap tidak menginjak tangga dan pion tidak dinaikan/dipindahkan, namun ketika 3 pertanyaan tersebut terpenuhi/jawabannya bernilai iya maka pion dianggap menginjak tangga dan pion dinaikan/dipindahkan ke posisi yang sudah ditentukan. Setelah 3

pertanyaan tersebut terpenuhi/jawabannya bernilai iya, sistem akan menjalankan perintah *play* untuk membunyikan efek naik, setelah itu sistem akan memindahkan pion dengan perintah *gridMove* ke posisi nilai *variable* PosisiMulaiSetelahNaikX dan PosisiMulaiSetelahNaikY yang sudah di set ketika pemasangan tangga. Dan *variable* PosisiP1x/PosisiP2x dan PosisiP1y/PosisiP2y juga di set menjadi nilai *variable* PosisiMulaiSetelahNaikX dan PosisiMulaiSetelahNaikY sebagai penanda bahwa pion berada di kotak yang yang berisi Ujung Tangga. Setelah itu sistem akan menunggu selama 0.5 detik. Setelah 0,5 detik, sistem akan menghapus ujung tangga, awalan tangga, dan animasi tangganya dengan perintah *destroy*, dan *set variable* pemilik tangga tersebut menjadi kosong, serta menjalankan efek suara tangga hancur.

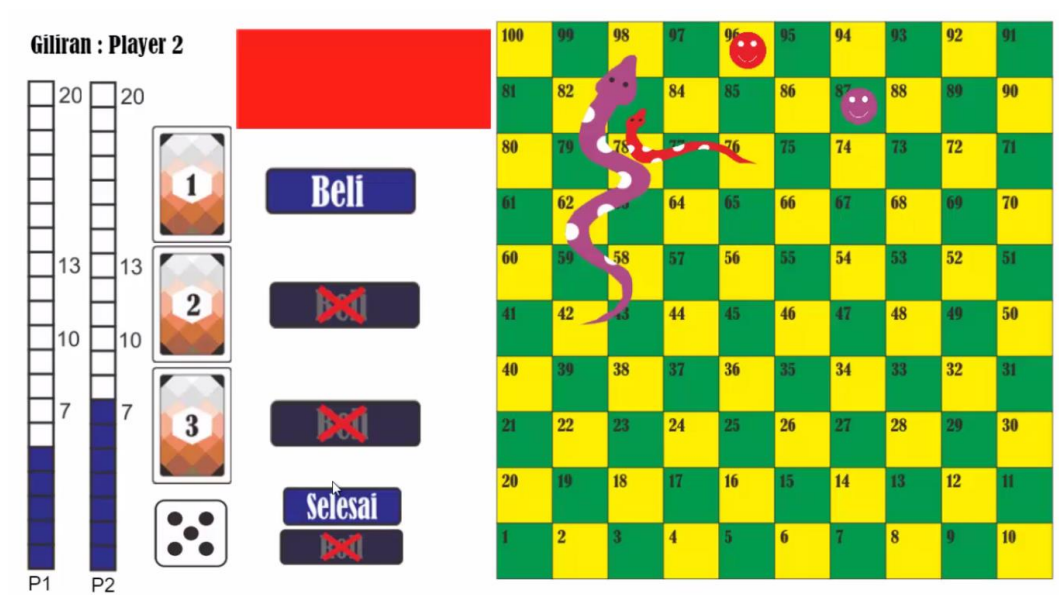
Tampilan event Pion Mengenai Tangga dapat dilihat pada gambar 4.106, 4.107, dan 4.108.



Gambar 4.106 Tampilan sebelum pion terkena tangga



Gambar 4.107 Tampilan pion menginjak awalan tangga



Gambar 4.108 Tampilan setelah pion naik akibat menginjak tangga

k. Event Ular Mati/Tangga Hancur karena Umur

Event ular mati/tangga hancur karena umur adalah event dimana objek ular/tangga menghilang dari papan karena umurnya sudah 0, event ini dijalankan setelah pemain menekan tombol selesai. Event ini berfungsi untuk menghilangkan objek ular/tangga dari papan karena umurnya sudah 0.

Condition Commands yang digunakan pada event ini adalah *Compare two values* dan *Is on layer*. *Action commands* pada event ini adalah *subtract from*, *play*, *set value*, dan *Destroy*. Dan *variable* yang digunakan pada event ini yaitu *umur*.

System	TanggaA1Y1.pemilik = "P2"	TanggaA1Y1	Subtract 1 from umur
		Add action	
System	TanggaA1Y2.pemilik = "P2"	TanggaA1Y2	Subtract 1 from umur
		Add action	
System	TanggaA1Ki2Y1.pemilik = "P2"	TanggaA1Ki2Y1	Subtract 1 from umur
		Add action	
System	TanggaA1Ki2Y2.pemilik = "P2"	TanggaA1Ki2Y2	Subtract 1 from umur
		Add action	
System	TanggaA1Ka2Y1.pemilik = "P2"	TanggaA1Ka2Y1	Subtract 1 from umur
		Add action	
System	TanggaA1Ka2Y2.pemilik = "P2"	TanggaA1Ka2Y2	Subtract 1 from umur
		Add action	
System	TanggaA2Y1.pemilik = "P2"	TanggaA2Y1	Subtract 1 from umur
		Add action	
System	TanggaA2Y2.pemilik = "P2"	TanggaA2Y2	Subtract 1 from umur
		Add action	
System	TanggaA2Ki1Y1.pemilik = "P2"	TanggaA2Ki1Y1	Subtract 1 from umur
		Add action	
System	TanggaA2Ki1Y2.pemilik = "P2"	TanggaA2Ki1Y2	Subtract 1 from umur
		Add action	
System	TanggaA2Ka1Y1.pemilik = "P2"	TanggaA2Ka1Y1	Subtract 1 from umur
		Add action	
System	TanggaA2Ka1Y2.pemilik = "P2"	TanggaA2Ka1Y2	Subtract 1 from umur
		Add action	

Gambar 4.109 Beberapa *source code* pengurangan umur tangga

System	UlarB1Y1.pemilik = "P1"	UlarB1Y1	Subtract 1 from umur
		Add action	
System	UlarB1Y2.pemilik = "P1"	UlarB1Y2	Subtract 1 from umur
		Add action	
System	UlarB1Ki2Y1.pemilik = "P1"	UlarB1Ki2Y1	Subtract 1 from umur
		Add action	
System	UlarB1Ki2Y2.pemilik = "P1"	UlarB1Ki2Y2	Subtract 1 from umur
		Add action	
System	UlarB1Ka2Y1.pemilik = "P1"	UlarB1Ka2Y1	Subtract 1 from umur
		Add action	
System	UlarB1Ka2Y2.pemilik = "P1"	UlarB1Ka2Y2	Subtract 1 from umur
		Add action	
System	UlarB2Y1.pemilik = "P1"	UlarB2Y1	Subtract 1 from umur
		Add action	
System	UlarB2Y2.pemilik = "P1"	UlarB2Y2	Subtract 1 from umur
		Add action	
System	UlarB2Ki1Y1.pemilik = "P1"	UlarB2Ki1Y1	Subtract 1 from umur
		Add action	
System	UlarB2Ki1Y2.pemilik = "P1"	UlarB2Ki1Y2	Subtract 1 from umur
		Add action	
System	UlarB2Ka1Y1.pemilik = "P1"	UlarB2Ka1Y1	Subtract 1 from umur
		Add action	
System	UlarB2Ka1Y2.pemilik = "P1"	UlarB2Ka1Y2	Subtract 1 from umur
		Add action	

Gambar 4.110 Beberapa *source code* pengurangan umur ular

Gambar 4.109 dan 4.110 menjelaskan *source code* setelah pemain selesai dengan gilirannya dan menekan tombol selesai, sistem akan langsung mengurangi 1 umur dari objek-objek milik pemain yang ada di papan permainan dengan perintah *subtract from*.

System	UlarB1Y1.umur ≤ 0	UlarB1Y1	Destroy
UlarB1Y1	Is on layer 1	UlarB1Y1Finish	Destroy
		UlarB1Y1	Set pemilik to "-"
		UlarLv1Ke1	Destroy
		Audio	Play UlarMati not looping at volume 0 dB (tag "UlarMati")
		Add action	
System	UlarB1Y2.umur ≤ 0	UlarB1Y2	Destroy
UlarB1Y2	Is on layer 1	UlarB1Y2Finish	Destroy
		UlarB1Y2	Set pemilik to "-"
		UlarLv1Ke2	Destroy
		Audio	Play UlarMati not looping at volume 0 dB (tag "UlarMati")
		Add action	
System	UlarB1Ki2Y1.umur ≤ 0	UlarB1Ki2Y1	Destroy
UlarB1Ki2Y1	Is on layer 1	UlarB1Ki2Y1Finish	Destroy
		UlarB1Ki2Y1	Set pemilik to "-"
		UlarLv1Ke3	Destroy
		Audio	Play UlarMati not looping at volume 0 dB (tag "UlarMati")
		Add action	
System	UlarB1Ki2Y2.umur ≤ 0	UlarB1Ki2Y2	Destroy
UlarB1Ki2Y2	Is on layer 1	UlarB1Ki2Y2Finish	Destroy
		UlarB1Ki2Y2	Set pemilik to "-"
		UlarLv1Ke4	Destroy
		Audio	Play UlarMati not looping at volume 0 dB (tag "UlarMati")
		Add action	
System	UlarB1Ka2Y1.umur ≤ 0	UlarB1Ka2Y1	Destroy
UlarB1Ka2Y1	Is on layer 1	UlarB1Ka2Y1Finish	Destroy
		UlarB1Ka2Y1	Set pemilik to "-"
		UlarLv1Ke5	Destroy

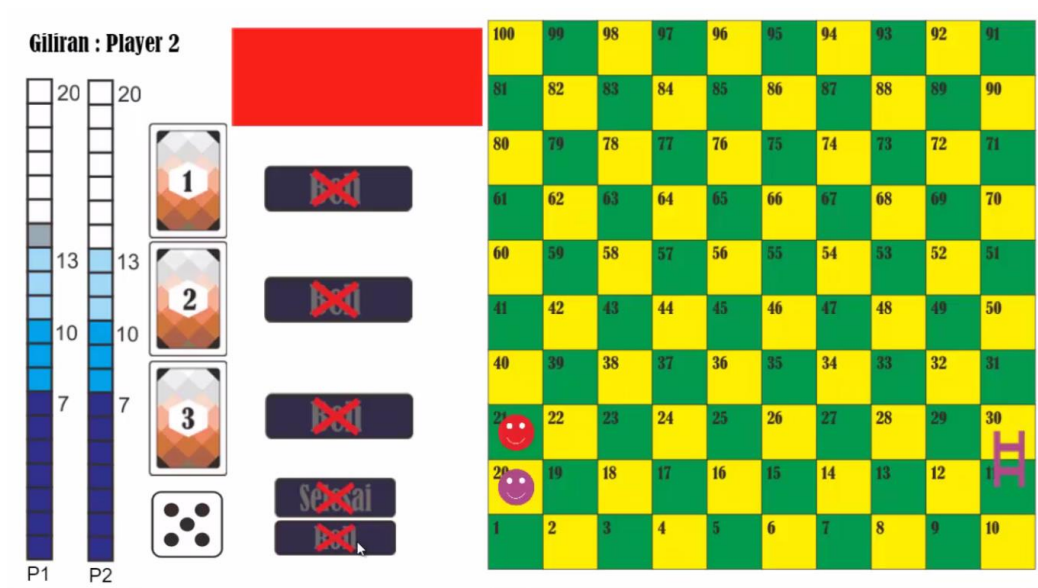
Gambar 4.111 Beberapa *source code* menghapus ular dari papan permainan karena umur

System	TanggaA1Y1.umur ≤ 0	TanggaA1Y1	Destroy
TanggaA1Y1	Is on layer 1	TanggaA1Y1Finish	Destroy
		TanggaLv1Ke1	Destroy
		TanggaA1Y1	Set pemilik to "-"
		Audio	Play TanggaHancur not looping at volume 0 dB (tag "TanggaHancur")
		Add action	
System	TanggaA1Y2.umur ≤ 0	TanggaA1Y2	Destroy
TanggaA1Y2	Is on layer 1	TanggaA1Y2Finish	Destroy
		TanggaLv1Ke2	Destroy
		TanggaA1Y2	Set pemilik to "-"
		Audio	Play TanggaHancur not looping at volume 0 dB (tag "TanggaHancur")
		Add action	
System	TanggaA1Ki2Y1.umur ≤ 0	TanggaA1Ki2Y1	Destroy
TanggaA1Ki2Y1	Is on layer 1	TanggaA1Ki2Y1Fini	Destroy
		TanggaA1Ki2Y1	Set pemilik to "-"
		TanggaLv1Ke3	Destroy
		Audio	Play TanggaHancur not looping at volume 0 dB (tag "TanggaHancur")
		Add action	
System	TanggaA1Ki2Y2.umur ≤ 0	TanggaA1Ki2Y2	Destroy
TanggaA1Ki2Y2	Is on layer 1	TanggaA1Ki2Y2Fini	Destroy
		TanggaA1Ki2Y2	Set pemilik to "-"
		TanggaLv1Ke4	Destroy
		Audio	Play TanggaHancur not looping at volume 0 dB (tag "TanggaHancur")
		Add action	

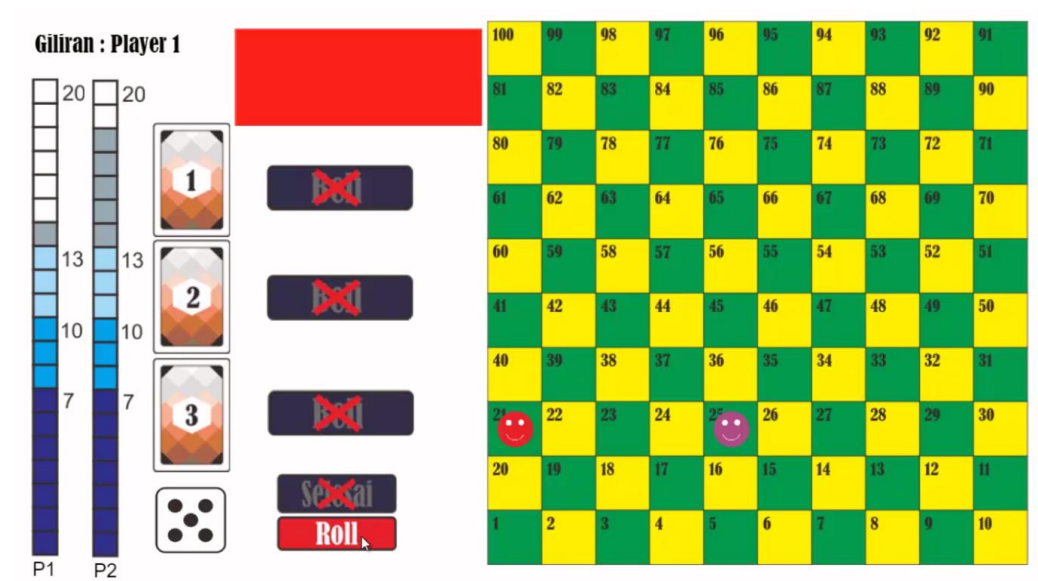
Gambar 4.112 Beberapa *source code* menghapus tangga dari papan permainan karena umur

Gambar 4.111 dan 4.112 menjelaskan *source code* ketika umur sama dengan 0, sistem akan menghapus objek dari papan dengan perintah *destroy*, sistem *set variable* pemilik menjadi kosong, dan sistem membunyikan efek menghilangnya objek.

Tampilan ular mati/tangga hancur karena umur dapat dilihat pada gambar 4.113 dan 4.114.



Gambar 4.113 Tampilan sebelum objek ular mati /tangga hancur karena umur

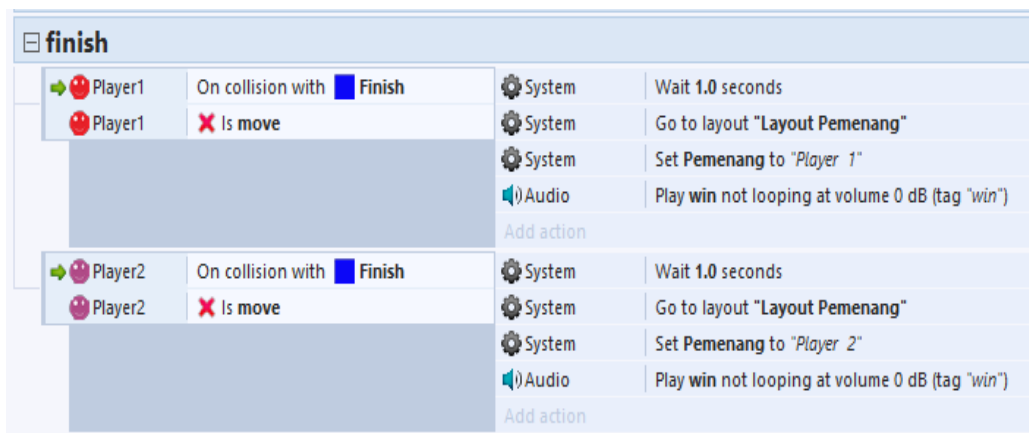


Gambar 4.114 Tampilan setelah objek ular mati /tangga hancur karena umur

1. Event Finish

Event finish adalah event dimana pion mencapai kotak yang ke 100 dengan tepat. Event ini berfungsi untuk menandakan bahwa permainan telah berakhir.

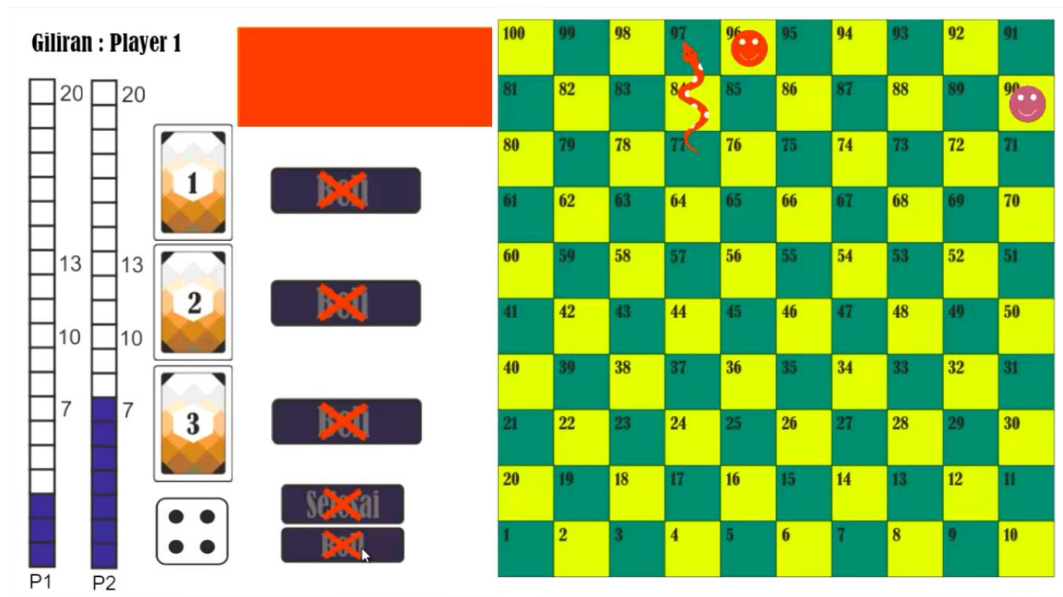
Condition Commands yang digunakan pada event ini adalah *On Collision with* dan *is boolean*. *Action commands* pada event ini adalah *wait*, *go to layout*, *set value*, dan *Play*. Dan ada 1 *variable* yang digunakan pada event ini yaitu Pemenang.



Gambar 4.115 Source code pada event finish

Gambar 4.115 menjelaskan *source code* ketika pemain mencapai *finish*, setelah event roll dadu pada saat permainan selesai. Pada gambar 4.78 sistem akan mengecek apakah pion pemain berada tepat di kotak ke 100 pada saat *boolean move* bernilai *false*. Ketika pion pemain berada tepat di kotak ke 100 dan *boolean move* bernilai *false*, maka sistem akan menjalankan *action commands wait* selama 1 detik agar ketika pion tepat di kotak ke 100 ada jeda untuk ke halaman selanjutnya. Setelah 1 detik sistem akan menampilkan *layout* pemenang dan sistem set *variable* pemenang menjadi pemenang permainan ini yang akan ditampilkan di *layout* pemenang. Setelah itu musik kemenangan diputar.

Tampilan event *finish* dapat dilihat pada gambar 4.116 dan 4.117.



Gambar 4.116 Tampilan sebelum finish



Gambar 4.117 Tampilan setelah Finish

4.3 Testing

Tahap *testing* adalah tahap melakukan pengujian terhadap permainan yang dibangun. Ada 3 kategori pengujian yang dilakukan dalam tahap ini. Pengujian yang pertama adalah pengujian fungsi aplikasi dengan menggunakan metode *blackbox testing*. Pengujian yang kedua adalah pengujian performa PC. Pengujian yang ketiga adalah pengujian portabilitas.

4.3.1 Pengujian fungsi

Pengujian fungsi adalah pengujian yang dilakukan untuk mengetahui apakah fungsi berjalan sesuai dengan yang diharapkan atau tidak. Pengujian ini menggunakan metode pengujian *black box*. pengujian *black box* adalah pengujian yang tidak perlu tahu apa yang sesungguhnya terjadi dalam sistem, yang diuji adalah masukan serta keluarannya. Dengan berbagai masukan yang diberikan, apakah sistem memberikan keluaran seperti yang diharapkan atau tidak.

a. Tujuan Pengujian

Pengujian dilakukan terhadap sistem aplikasi permainan ular tangga bergenre strategi ini bertujuan untuk menjamin bahwa sistem yang dibangun memiliki kualitas yang baik dan bebas dari *bug*.

b. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan hasil dari beberapa kriteria yang terpenuhi, kriteria pengujian dibagi menjadi 3 bagian yaitu pengujian fungsi aplikasi, pengujian fungsi objek ular, dan pengujian fungsi objek tangga. Kriteria pengujian dan hasil yang diharapkan dapat dilihat pada tabel 4.4, 4.5, dan 4.6.

Tabel 4.4 Kriteria dan hasil yang diharapkan pada pengujian fungsi aplikasi

No.	Kasus Yang Diuji	Masukan	Hasil Yang Diharapkan
1	Tombol Bermain	User menekan tombol bermain	Menampilkan <i>layout preparation game</i>
2	Tombol Cara Bermain	User menekan tombol cara bermain	Menampilkan <i>layout</i> tentang permainan
3	Tombol Keluar (<i>layout</i> utama)	User menekan tombol keluar	Keluar dari aplikasi
4	Tombol selanjutnya	<i>User</i> menekan	Menampilkan <i>layout</i>

No.	Kasus Yang Diuji	Masukan	Hasil Yang Diharapkan
	<i>(layout tentang permainan)</i>	tombol selanjutnya	cara bermain
5	Tombol sebelumnya <i>(layout cara bermain)</i>	<i>User</i> menekan tombol sebelumnya	Menampilkan <i>layout</i> tentang permainan
6	Tombol selesai <i>(layout cara bermain)</i>	<i>User</i> menekan tombol selesai	Menampilkan <i>layout</i> utama/home
7	Tombol <i>play</i> video	<i>User</i> menekan tombol <i>play</i> video	Sistem memutar video
8	Tombol <i>pause</i> video	<i>User</i> menekan tombol <i>pause</i> video	Sistem mem- <i>pause</i> video
9	Tombol <i>reload</i> video	<i>User</i> menekan tombol <i>reload</i>	Sistem memutar ulang video
10	Video cara bermain	<i>Sistem</i> menampilkan halaman cara bermain	Video memutar secara otomatis
11	Tombol roll <i>(layout preparation game)</i>	<i>User</i> menekan tombol roll	menampilkan <i>text</i> yang mendapat giliran pertama dan membuka tombol <i>play</i> di <i>layout preparation game</i>
12	Tombol <i>play</i> <i>(layout preparation game)</i>	<i>User</i> menekan tombol <i>play</i>	Menampilkan <i>layout</i> permainan dan menampilkan giliran

No.	Kasus Yang Diuji	Masukan	Hasil Yang Diharapkan
			sesuai urutan.
13	Tombol roll (<i>layout</i> permainan)	<i>User</i> menekan tombol roll	Menampilkan pengacakan dadu dan menampilkan angka dadu yang didapat pemain
14	Pion melangkah	Sistem menampilkan angka dadu	Pion melangkah sejumlah angka dadu yang didapat.
15	Poin bertambah	Pion melangkah	Poin bertambah sesuai jumlah langkah pion
16	Tombol selesai (<i>layout</i> permainan)	<i>User</i> menekan tombol selesai	Giliran berganti dan tombol roll berganti sesuai yang mendapat giliran
17	Membuka tombol beli kartu level 1 ketika poin sama dengan 7	Poin pemain sama dengan 7	tombol beli kartu level 1 terbuka
18	Membuka tombol beli kartu level 2 ketika poin sama dengan 10	Poin pemain sama dengan 10	tombol beli kartu level 2 terbuka
19	Membuka tombol beli kartu level 3 ketika poin sama dengan 13	Poin pemain sama dengan 13	tombol beli kartu level 3 terbuka

No.	Kasus Yang Diuji	Masukan	Hasil Yang Diharapkan
20	tombol beli kartu level 1	<i>User</i> menekan tombol beli kartu level 1	Menampilkan kartu level 1 yang didapat , mengurangi poin sesuai harga kartu level 1, dan menampilkan pilihan untuk dijual atau dipasang
21	tombol beli kartu level 2	<i>User</i> menekan tombol beli kartu level 2	Menampilkan kartu level 2 yang didapat, mengurangi poin sesuai harga kartu level 1, dan menampilkan pilihan untuk dijual atau dipasang
22	tombol beli kartu level 3	<i>User</i> menekan tombol beli kartu level 3	Menampilkan kartu level 3 yang didapat, mengurangi poin sesuai harga kartu level 1, dan menampilkan pilihan untuk dijual atau dipasang
23	tombol jual kartu level 1	<i>User</i> menekan tombol jual kartu	Menampilkan poin yang sudah ditambahkan sesuai

No.	Kasus Yang Diuji	Masukan	Hasil Yang Diharapkan
		level 1	harga jual kartu level 1
24	tombol jual kartu level 2	<i>User</i> menekan tombol jual kartu level 2	Menampilkan poin yang sudah ditambahkan sesuai harga jual kartu level 2
25	tombol jual kartu level 3	<i>User</i> menekan tombol jual kartu level 3	Menampilkan poin yang sudah ditambahkan sesuai harga jual kartu level 3
26	tombol pasang kartu level 1	<i>User</i> menekan tombol pasang kartu level 1	Seluruh kotak pada papan permainan bisa dipilih/dapat ditekan
27	tombol pasang kartu level 2	<i>User</i> menekan tombol pasang kartu level 2	Seluruh kotak pada papan permainan bisa dipilih/dapat ditekan
28	tombol pasang kartu level 3	<i>User</i> menekan tombol pasang kartu level 3	Seluruh kotak pada papan permainan dapat dipilih/ditekan
29	Pemeriksaan kotak yang dipilih	<i>User</i> menekan salah satu kotak pada papan permainan	Menampilkan objek kartu/ menampilkan peringatan
30	kotak ke 100 pada papan permainan dipilih/ditekan saat	<i>User</i> menekan kotak yang ke 100 pada papan permainan	menampilkan peringatan

No.	Kasus Yang Diuji	Masukan	Hasil Yang Diharapkan
	pemain ingin pasang objek		
31	Finish/Selesai	Pion berhenti tepat di kotak ke 100	Menampilkan <i>layout</i> pemenang
32	tombol main lagi	User menekan tombol main lagi	Menampilkan <i>layout preparation game</i>
33	Tombol Keluar (<i>layout</i> pemenang)	User menekan tombol keluar	Keluar dari aplikasi

Tabel 4.5 Kriteria dan hasil yang diharapkan pada pengujian fungsi objek ular

Kasus yang diuji	Hasil yang diharapkan
Pasang kartu	Menampilkan objek ular sesuai ketentuan kartu
Ekor keluar arena	Menampilkan peringatan
Ekor Mengenai objek lain	Menampilkan peringatan
Pion menginjak ular lawan	Pion dipindahkan ke ekor ular dan ular menghilang dari papan permainan
Pion menginjak ular sendiri	Pion tidak dipindahkan
Umur ular = 0	Ular menghilang dari papan permainan

Tabel 4.6 Kriteria dan hasil yang diharapkan pada pengujian fungsi objek tangga

Kasus yang diuji	Hasil yang diharapkan
Pasang kartu	Menampilkan animasi tangga sesuai ketentuan kartu
Ujung tangga keluar arena	Menampilkan peringatan
Ujung tangga mengenai objek lain	Menampilkan peringatan
Ujung tangga berada di kotak 100	Menampilkan peringatan
Pion menginjak awalan tangga sendiri	Pion dipindahkan ke ujung tangga dan tangga menghilang dari papan permainan
Pion menginjak awalan tangga lawan	Pion tidak dipindahkan
Umur tangga = 0	Tangga menghilang dari papan permainan

c. Lingkungan Pengujian

Lingkungan pengujian aplikasi permainan ular tangga bergenre strategi dilakukan pada PC Laptop TOSHIBA satellite C50-A536 dengan Spesifikasi sebagai berikut :

1. Processor : Intel(R)Core(TM)i3-3110U 2.4GHz
2. Memory : 4 GB
3. HDD : 320 GB
4. VGA : 512 MB
5. OS : Windows 7

d. Prosedur Pengujian

Prosedur pengujian fungsi pada permainan ular tangga bergenre strategi menggunakan sebuah data masukan dari pemain. Pemain melakukan pengujian langsung terhadap kriteria yang diujikan dengan cara memainkan permainan ular tangga bergenre strategi dan memeriksa apakah fungsi dari setiap kriteria pengujian sesuai atau tidak. Setiap kriteria pengujian dilakukan 3 kali uji.

e. Hasil dan Pembahasan Pengujian

1. Hasil Pengujian

Hasil dari pengujian terhadap pengujian fungsi aplikasi, pengujian fungsi objek ular, dan pengujian fungsi objek tangga dapat dilihat pada tabel 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, dan 4.13. Untuk hasil pengujian fungsi objek ular dibagi menjadi 3 tabel berdasarkan levelnya. Begitu juga hasil pengujian fungsi objek tangga dibagi menjadi 3 tabel berdasarkan levelnya juga.

Tabel 4.7 Hasil Pengujian fungsi aplikasi

No.	Kasus Yang Diuji	Hasil Yang Diharapkan	Hasil yang Diterima	Status
1	Tombol Bermain	Menampilkan <i>layout preparation game</i>	Sistem menampilkan <i>layout preparation game</i>	Berhasil
2	Tombol Cara Bermain	Menampilkan <i>layout</i> tentang permainan	Sistem menampilkan <i>layout</i> tentang permainan	Berhasil
3	Tombol Keluar (<i>layout</i> utama)	Keluar dari aplikasi	Sistem keluar dari aplikasi	Berhasil
4	Tombol selanjutnya	Menampilkan <i>layout</i> cara bermain	Sistem menampilkan <i>layout</i> cara bermain	Berhasil

No.	Kasus Yang Diuji	Hasil Yang Diharapkan	Hasil yang Diterima	Status
	<i>(layout</i> tentang permainan)			
5	Tombol sebelumnya (<i>layout</i> cara bermain)	Menampilkan <i>layout</i> tentang permainan	Sistem menampilkan <i>layout</i> tentang permainan	Berhasil
6	Tombol selesai (<i>layout</i> cara bermain)	Menampilkan <i>layout</i> utama/home	Sistem menampilkan <i>layout</i> utama/home	Berhasil
7	Tombol <i>play</i> video	Sistem memutar video	Sistem memutar video	Behasil
8	Tombol <i>pause</i> video	Sistem mem- <i>pause</i> video	Sistem mem- <i>pause</i> video	Berhasil
9	Tombol <i>reload</i> video	Sistem memutar ulang video	Sistem memutar ulang video	Berhasil
10	Video cara bermain	Video memutar secara otomatis	Sistem memutar video secara otomatis	Berhasil
11	Tombol roll (<i>layout preparation game</i>)	menampilkan <i>text</i> yang mendapat giliran pertama dan membuka tombol play di <i>layout preparatin game</i>	Sistem menampilkan <i>text</i> yang mendapat giliran pertama dan sistem membuka tombol play di <i>layout preparation</i>	Berhasil

No.	Kasus Yang Diuji	Hasil Yang Diharapkan	Hasil yang Diterima	Status
			<i>game</i>	
12	Tombol play (<i>layout preparation game</i>)	Menampilkan <i>layout</i> permainan dan menampilkan giliran sesuai urutan.	Sistem Menampilkan <i>layout</i> permainan dan menampilkan giliran sesuai urutan.	Berhasil
13	Tombol roll (<i>layout permainan</i>)	Menampilkan pengacakan dadu dan menampilkan angka dadu yang didapat pemain	Sistem Menampilkan <i>layout</i> permainan dan menampilkan giliran sesuai urutan.	Berhasil
14	Pion melangkah	Pion melangkah sejumlah angka dadu yang didapat.	Sistem menjalankan pion sesuai angka dadu yang didapat	Berhasil
15	Poin bertambah	Poin bertambah sesuai jumlah langka pion	Sistem Menambahkan poin sesuai jumlah langkah pion	Berhasil
16	Tombol selesai (<i>layout permainan</i>)	Giliran berganti dan tombol roll berganti sesuai yang mendapat giliran	Sistem menampilkan Giliran berganti dan tombol roll berganti sesuai yang mendapat giliran	Berhasil
17	Membuka tombol beli	tombol beli kartu level 1 terbuka	Sistem mebuca tombol beli kartu	Berhasil

No.	Kasus Yang Diuji	Hasil Yang Diharapkan	Hasil yang Diterima	Status
	kartu level 1 ketika poin sama dengan 7		level 1 ketika poin pemain sudah = 7 atau lebih	
18	Membuka tombol beli kartu level 2 ketika poin sama dengan 10	tombol beli kartu level 2 terbuka	Sistem membuka tombol beli kartu level 2 ketika poin pemain sudah = 10 atau lebih	Berhasil
19	Membuka tombol beli kartu level 3 ketika poin sama dengan 13	tombol beli kartu level 3 terbuka	Sistem membuka tombol beli kartu level 3 ketika poin pemain sudah = 13 atau lebih	Berhasil
20	tombol beli kartu level 1	Menampilkan kartu level 1 yang didapat, mengurangi poin sesuai harga kartu level 1, dan menampilkan pilihan untuk dijual atau dipasang	Sistem Menampilkan kartu level 1 yang didapat pemain, mengurangi poin sejumlah 7, dan menampilkan pilihan untuk dijual atau dipasang	Berhasil
21	tombol beli kartu level 2	Menampilkan kartu level 2 yang	Sistem Menampilkan kartu level 2 yang	Berhasil

No.	Kasus Yang Diuji	Hasil Yang Diharapkan	Hasil yang Diterima	Status
		didapat, mengurangi poin sesuai harga kartu level 1, dan menampilkan pilihan untuk dijual atau dipasang	didapat pemain, mengurangi poin sejumlah 10, dan menampilkan pilihan untuk dijual atau dipasang	
22	tombol beli kartu level 3	Menampilkan kartu level 3 yang didapat, mengurangi poin sesuai harga kartu level 1, dan menampilkan pilihan untuk dijual atau dipasang	Sistem Menampilkan kartu level 2 yang didapat pemain, mengurangi poin sejumlah 13, dan menampilkan pilihan untuk dijual atau dipasang	Berhasil
23	tombol jual kartu level 1	Menampilkan poin yang sudah ditambahkan sesuai harga jual kartu level 1	Sistem menambahkan poin sejumlah 4 poin, dan menampilkan poin yang sudah ditambahkan	Berhasil
24	tombol jual kartu level 2	Menampilkan poin yang sudah ditambahkan sesuai harga jual kartu	Sistem menambahkan poin sejumlah 6 poin, dan menampilkan poin yang sudah	Berhasil

No.	Kasus Yang Diuji	Hasil Yang Diharapkan	Hasil yang Diterima	Status
		level 2	ditambahkan	
25	tombol jual kartu level 3	Menampilkan poin yang sudah ditambahkan sesuai harga jual kartu level 3	Sistem menambahkan poin sejumlah 8 poin, dan menampilkan poin yang sudah ditambahkan	Berhasil
26	tombol pasang kartu level 1	Seluruh kotak pada papan permainan bisa dipilih/dapat ditekan	Sistem membuka seluruh kotak pada papan dan semua kotak bisa dipilih	Berhasil
27	tombol pasang kartu level 2	Seluruh kotak pada papan permainan bisa dipilih/dapat ditekan	Sistem membuka seluruh kotak pada papan dan semua kotak bisa dipilih	Berhasil
28	tombol pasang kartu level 3	Seluruh kotak pada papan permainan dapat dipilih/ditekan	Sistem membuka seluruh kotak pada papan dan semua kotak bisa dipilih	Berhasil
29	Pemeriksaan kotak yang dipilih	Menampilkan objek kartu/ menampilkan peringatan	Sistem Menampilkan objek kartu/ menampilkan peringatan	Berhasil
30	kotak ke 100 pada papan	menampilkan peringatan	Sistem menampilkan tidak boleh	Berhasil

No.	Kasus Yang Diuji	Hasil Yang Diharapkan	Hasil yang Diterima	Status
	permainan dipilih/ditekan saat pemain ingin pasang objek		memasang apapun di 100	
31	Finish/Selesai	Menampilkan <i>layout</i> pemenang	Sistem menampilkan <i>layout</i> pemenang	Berhasil
32	tombol main lagi	Menampilkan <i>layout preparation game</i>	Sistem menampilkan <i>layout preparation game</i>	Berhasil
33	Tombol Keluar (<i>layout</i> pemenang)	Keluar dari aplikasi	Sistem keluar dari aplikasi	Berhasil

Tabel 4.8 Hasil Pengujian fungsi Ular level 1

Ular Kasus	Level 1 yang ke-1	Level 1 yang ke-2	Level 1 yang ke-3	Level 1 yang ke-4	Level 1 yang ke-5	Level 1 yang ke-6	Level 1 yang ke-7	Level 1 yang ke-8	Level 1 yang ke-9	Level 1 yang ke-10	Level 1 yang ke-11	Level 1 yang ke-12
Pasang kartu	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ekor keluar arena	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ekor Mengenai objek lain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak ular lawan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak ular sendiri	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Umur ular = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Keterangan

✓ = Berhasil

X = Tidak Berhasil

Tabel 4.9 Hasil Pengujian fungsi Ular level 2

Ular	Level 2 yang ke-1	Level 2 yang ke-2	Level 2 yang ke-3	Level 2 yang ke-4	Level 2 yang ke-5	Level 2 yang ke-6	Level 2 yang ke-7	Level 2 yang ke-8
Kasus								
Pasang kartu	✓	✓	✓	✓	✓	✓	✓	✓
Ekor keluar arena	✓	✓	✓	✓	✓	✓	✓	✓
Ekor Mengenai objek lain	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak ular lawan	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak ular sendiri	✓	✓	✓	✓	✓	✓	✓	✓
Umur ular = 0	✓	✓	✓	✓	✓	✓	✓	✓

Keterangan

✓ = Berhasil

X = Tidak Berhasil

Tabel 4.10 Hasil Pengujian fungsi Ular level 3

Ular	Level 3 yang ke-1	Level 3 yang ke-2	Level 3 yang ke-3	Level 3 yang ke-4	Level 3 yang ke-5	Level 3 yang ke-6	Level 3 yang ke-7	Level 3 yang ke-8	Level 3 yang ke-9
Kasus									
Pasang kartu	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ekor keluar arena	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ekor Mengenai objek lain	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak ular lawan	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak ular sendiri	✓	✓	✓	✓	✓	✓	✓	✓	✓
Umur ular = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓

Keterangan

✓ = Berhasil

X = Tidak Berhasil

Tabel 4.11 Hasil Pengujian fungsi Tangga level 1

Tangga	Level 1 yang ke-1	Level 1 yang ke-2	Level 1 yang ke-3	Level 1 yang ke-4	Level 1 yang ke-5	Level 1 yang ke-6	Level 1 yang ke-7	Level 1 yang ke-8	Level 1 yang ke-9	Level 1 yang ke-10	Level 1 yang ke-11	Level 1 yang ke-12
Kasus												
Pasang kartu	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ujung tangga keluar arena	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ujung tangga Mengenai objek lain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ujung tangga berada di kotak 100	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak tangga lawan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak tangga sendiri	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Umur tangga = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Keterangan

✓ = Berhasil

X = Tidak Berhasil

Tabel 4.12 Hasil Pengujian fungsi Tangga level 2

Tangga	Level 2 yang ke-1	Level 2 yang ke-2	Level 2 yang ke-3	Level 2 yang ke-4	Level 2 yang ke-5	Level 2 yang ke-6	Level 2 yang ke-7	Level 2 yang ke-8
Kasus								
Pasang kartu	✓	✓	✓	✓	✓	✓	✓	✓
Ujung tangga keluar arena	✓	✓	✓	✓	✓	✓	✓	✓
Ujung tangga Mengenai objek lain	✓	✓	✓	✓	✓	✓	✓	✓
Ujung tangga berada di kotak 100	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak tangga lawan	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak tangga sendiri	✓	✓	✓	✓	✓	✓	✓	✓
Umur tangga = 0	✓	✓	✓	✓	✓	✓	✓	✓

Keterangan

✓ = Berhasil

X = Tidak Berhasil

Tabel 4.13 Hasil Pengujian fungsi Tangga level 3

Tangga	Level 3 yang ke-1	Level 3 yang ke-2	Level 3 yang ke-3	Level 3 yang ke-4	Level 3 yang ke-5	Level 3 yang ke-6	Level 3 yang ke-7	Level 3 yang ke-8	Level 3 yang ke-9
Kasus									
Pasang kartu	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ujung tangga keluar arena	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ujung tangga Mengenai objek lain	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ujung tangga berada di kotak 100	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak tangga lawan	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pion menginjak tangga sendiri	✓	✓	✓	✓	✓	✓	✓	✓	✓
Umur tangga = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓

Keterangan

✓ = Berhasil

X = Tidak Berhasil

2. Pembahasan Pengujian

Pada pengujian fungsi aplikasi, dari keseluruhan poin yang diujikan semuanya dapat berjalan atau memenuhi kriteria yang diharapkan seperti yang terlihat pada tabel 4.4. Pada pengujian objek ular, dari keseluruhan objek ular yang berjumlah 29 yang terbagi menjadi 12 objek ular level 1, 8 objek ular level 2, dan 9 objek ular level 3, semua poin dari pengujian setiap objek ular menampilkan hasil sesuai dengan apa yang diharapkan, seperti yang terlihat pada tabel 4.5, 4.6, dan 4.7. Begitu juga pada pengujian objek tangga, dari keseluruhan objek tangga yang berjumlah 29 yang terbagi menjadi 12 objek tangga level 1, 8 objek tangga level 2, dan 9 objek tangga level 3, semua poin dari pengujian setiap objek tangga menampilkan hasil sesuai dengan apa yang diharapkan seperti yang terlihat pada tabel 4.8, 4.9, dan 4.10.

f. Kesimpulan Pengujian

Dari hasil pengujian yang telah dilakukan, permainan ular tangga bergenre strategi yang dibuat telah dapat digunakan dengan baik. Semua tombol dan fungsi telah berjalan dengan baik sehingga dapat digunakan oleh pemain.

4.3.2 Pengujian Performa

Pengujian performa adalah pengujian untuk menguji performa PC/Laptop dapat berjalan tanpa masalah saat memainkan permainan ular tangga bergenre strategi. Pengujian dilakukan pada hardware Memory(RAM) karena menurut Arfanizar Fathurochman [24] RAM adalah salah satu komponen hardware yang penting dalam PC/laptop *gaming*.

a. Tujuan Pengujian

Pengujian dilakukan terhadap sistem aplikasi permainan ular tangga bergenre strategi ini bertujuan untuk mengetahui sejauh mana performa PC/Laptop dapat berjalan saat memainkan permainan ular tangga bergenre

strategi. Diharapkan aplikasi ini dapat berjalan tanpa masalah dengan RAM yang berbeda-beda.

b. Lingkungan Pengujian

Menurut survei perusahaan analisis web yaitu StatCounter [25], windows 10 adalah OS Dekstop yang populer pada saat ini. Oleh karena itu pengujian dilakukan pada OS windows 10 dengan spesifikasi RAM yang berbeda-beda. Menurut Microsoft [26] spesifikasi RAM minimal windows 10 adalah 2 GB, maka pengujian performa dimulai dari RAM 2 GB sampai RAM yang 8 GB. Untuk kapasitas RAM yang diujikan yaitu:

1. 2 PC/Laptop dengan kapasitas RAM 2 GB
2. 2 PC/Laptop dengan kapasitas RAM 4 GB
3. 2 PC/Laptop dengan kapasitas RAM 8 GB

c. Prosedur Pengujian

Prosedur pengujian performa pada permainan ini adalah dengan membandingkan permainan ular tangga bergenre strategi dengan permainan sejenis yang memiliki elemen dan fitur yang hampir sama. Permainan tersebut adalah permainan *monopoly Dekstop/PC* yang dibuat oleh perusahaan EA (Electronic Art Inc.) dan Hasbro Inc. [27]. Sebelum memulai pengujian, setiap PC/laptop yang diuji dicatat terlebih dahulu penggunaan RAM-nya untuk mengukur antara kondisi awal dan kondisi pada saat dimainkan. Permainan dimainkan selama 20 menit pada masing-masing PC/Laptop dan setiap 5 menit sekali akan diukur kenaikannya.

d. Hasil dan Pembahasan Pengujian

1. Hasil

Hasil dari pengujian perforam dapat dilihat pada tabel 4.14. hasil didapatkan dari kondisi yang dipantau dikurangi dengan kondisi awal.

2. Pembahasan

Dari hasil pengujian seperti yang terlihat pada tabel 4.11, pada PC dengan RAM 2 GB untuk pemakaian RAM saat memainkan permainan ular tangga bergenre strategi sekitar 0,4 sampai 0,6 GB dengan rata-rata pemakaian adalah 0,525 GB. Sedangkan untuk pemakaian RAM pada saat

memainkan permainan *monopoly* PC sekitar 0,5 sampai 0,6 GB dengan rata-rata pemakaian PC 1 dengan 0,571 GB dan PC 4 dengan 0,55 GB. Saat memainkan 2 permainan ini, PC dengan RAM 2 GB agak sedikit *lag* dengan keterlambatan animasi sekitar 0,2 detik tetapi masih tahap wajar karena masih bisa dimainkan dengan baik.

Tabel 4.14 Hasil Pengujian Performa

Device	Memory (RAM)	Game	Kenaikan RAM (GB) di 5 menit ke-				Rata-rata (GB)
			1	2	3	4	
PC 1	2 GB	Ular Tangga	0,6	0,6	0,4	0,5	0,525
		<i>Monopoly</i> PC	0,5	0,6	0,6	0,6	0,575
PC 2	2 GB	Ular Tangga	0,6	0,6	0,5	0,4	0,525
		<i>Monopoly</i> PC	0,5	0,5	0,6	0,6	0,55
PC 3	4 GB	Ular Tangga	0,3	0,2	0,1	0,1	0,175
		<i>Monopoly</i> PC	0,2	0,3	0,3	0,2	0,25
PC 4	4 GB	Ular Tangga	0,2	0,3	0,3	0,3	0,275
		<i>Monopoly</i> PC	0,3	0,3	0,3	0,3	0,3
PC 5	8 GB	Ular Tangga	0,7	0,8	0,8	0,6	0,725
		<i>Monopoly</i> PC	0,6	0,5	0,6	0,7	0,6
PC 6	8 GB	Ular Tangga	0,5	0,4	0,5	0,5	0,475
		<i>Monopoly</i> PC	0,5	0,5	0,4	0,4	0,45

Pada PC dengan RAM 4 GB untuk pemakaian RAM saat memainkan permainan ular tangga bergenre strategi ini sekitar 0,2 sampai 0,3 GB dengan rata-rata pemakaian PC 3 dengan 0,175 GB dan PC 4 dengan 0,275 GB. Sedangkan untuk pemakaian RAM saat memainkan permainan *monopoly* PC sekitar 0,2 sampai 0,3 GB dengan rata-rata pemakaian PC 3 dengan 0,25 GB dan PC 4 dengan 0,3 GB. Saat memainkan

2 permainan ini pada PC dengan RAM 4 GB, permainan lancar dan tidak terjadi kendala apapun.

Pada PC dengan RAM 8 GB untuk pemakaian RAM saat memainkan permainan ular tangga bergenre strategi ini sekitar 0,4 sampai 0,7 GB dengan rata-rata pemakaian PC 5 dengan 0,725 GB dan PC 6 dengan 0,475 GB. Sedangkan untuk pemakaian RAM saat memainkan permainan *Monopoly* PC sekitar 0,4 sampai 0,7 GB dengan rata-rata pemakaian PC 5 dengan 0,6 GB dan PC 6 dengan 0,45 GB. Saat memainkan 2 permainan ini pada PC dengan RAM 8 GB, permainan lancar dan tidak terjadi kendala apapun.

e. Kesimpulan Pengujian

Secara keseluruhan hasil dari pengujian cukup memuaskan karena permainan ular tangga bergenre strategi berjalan stabil. Dengan konsumsi RAM dari 0,2 sampai 0,7 GB permainan ular tangga bergenre strategi untuk pemakaian RAM-nya masih normal karena kurang lebih sama dengan pemakaian permainan *monopoly* PC. Diharapkan untuk RAM 16 dan seterusnya lebih baik dalam memainkan permainan ular tangga bergenre strategi ini karena kapasitas RAM-nya lebih besar.

4.3.3 Pengujian Portabilitas Permainan

Pengujian Portabilitas adalah pengujian untuk mengetahui apakah aplikasi dapat berjalan dengan baik atau tidak pada *Operating System*(OS) windows yang berbeda-beda.

a. Tujuan Pengujian

Pengujian dilakukan terhadap sistem aplikasi permainan ular tangga bergenre strategi ini bertujuan untuk memastikan bahwa aplikasi permainan dapat berjalan pada OS windows yang berbeda-beda.

b. Lingkungan Pengujian

Menurut survei perusahaan analisis web yaitu StatCounter [25], OS windows yang populer pada saat ini adalah Windows 10, windows 7 dan Windows 8.1 oleh karena itu pengujian ini dilakukan pada OS windows 7, 8.1, dan 10.

c. Prosedur Pengujian

Prosedur pengujian portabilitas pada permainan ular tangga bergenre strategi yaitu memainkan permainan selama 20 menit dengan menggunakan 6 PC/Laptop dengan OS windows yang berbeda.

d. Hasil dan Pembahasan Pengujian

1. Hasil Pengujian

Hasil dari pengujian portabilitas OS dapat dilihat pada tabel 4.15.

Tabel 4.15 Hasil pengujian portabilitas OS permainan

No.	PC/Laptop	OS	Hasil Pengujian
1	PC-A	Windows 7	Berhasil
2	PC-B	Windows 7	Berhasil
3	PC-C	Windows 8.1	Berhasil
4	PC-D	Windows 8.1	Berhasil
5	PC-E	Windows 10	Berhasil
6	PC-F	Windows 10	Berhasil

2. Pembahasan

Pada pengujian portabilitas seperti yang terlihat pada tabel 4.11, pengujian dimulai pada PC-A dengan OS window 7. Pada pengujian pertama ini permainan diinstal dan dimainkan selama 20 menit. Dari awal instal dan sampai 20 menit berlalu, permainan dapat berjalan dengan baik. Begitu juga dengan pengujian yang kedua pada PC-B yang menggunakan OS Windows 7, pengujian yang ketiga pada PC-C yang menggunakan OS Windows 8.1, pengujian yang keempat pada PC-D yang menggunakan OS

Windows 8.1, pengujian yang kelima pada PC-E yang menggunakan OS Windows 10 dan sampai pengujian yang keenam pada PC-F yang menggunakan OS Windows 10, permainan dapat berjalan dengan baik sampai permainan dimainkan selama 20 menit.

e. Kesimpulan Pengujian

Dari hasil pengujian yang telah dilakukan, permainan ular tangga bergenre strategi yang dibuat telah dapat dimainkan pada OS windows yang berbeda-beda. Tetapi perlu diketahui bahwa untuk pengujian portabilitas OS windows yang berbeda-beda, spesifikasi hardware pada PC/Laptop yang digunakan untuk pengujian berbeda-beda.